Simulation of a Novel Electromechanical Engine Valve Drive to Quantify Performance Gains in Fuel Consumption

by

Justin Miller

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of
Master of Science
at the
MASSACHUSETTS INSTITUE OF TECHNOLOGY
June 2011

Author..
Department of Mechanical Engineering
May 18, 2011

Certified by..
John G. Kassakian
Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Certified by..
Wai K. Cheng
Professor, Mechanical Engineering
Thesis Supervisor

Accepted by..
David E. Hardt
Chairman, Committee on Graduate Students

Simulation of a Novel Electromechanical Engine Valve Drive to Quantify Performance Gains in Fuel Consumption

by

Justin Miller

Submitted to the Department of Mechanical Engineering on June 2011, in partial fulfillment of the requirements for the degree of
Master of Science

Abstract

This thesis desribes the modeling and simulation of a novel electromechanical valve drive known as the MIT EMV. This valve drive allows an engine to achieve variable valve timing which has been shown to produce improvements in engine fuel efficiency. To test this improvement, a reference engine model with fixed valve timing was obtained from the engine simulation software package WAVE® by Ricardo. A model of the MIT EMV was generated based on the details of the physical actuator, and it was incorporated into the WAVE® engine model. An interface between MATLAB® and WAVE® was developed for simulating the actuator at desired engine speeds and loads. Specific test points were chosen based on corporate operating points and operating points that were used to test the BMW Valvetronic actuator. Through simulation, it was determined that the MIT EMV can provide a reduction of approximately 10% in fuel consumption at the corporate operating points when compared to the reference engine model. The drive was also able to achieve performance gains similar to the BMW Valvetronic actuator, showing that it is able to compete with other actuators on the market even without variable lift capabilities.

Thesis Committee

Prof. John G. Kassakian (Supervisor)
Professor of Electrical Engineering and Computer Science

Prof. Wai K. Cheng (Thesis reader)
Professor of Mechanical Engineering

Acknowledgements

I would like to thank my adviser Professor John Kassakian for his advice, guidance, feedback, and funding. Without him, this project would not have been possible. I would also like to thank Professor Wai K. Cheng for agreeing to be a thesis reader. His support in providing software through the Sloan Automotive Laboratory and his knowledge of internal combustion engines has been incredibly helpful. Dr. Tom Keim was also instrumental in getting my project started and refined during my first two semesters, so I would like to thank him for his help as well.

I would like acknowledge Mark Subramaniam, manager at FEV Inc, who provided advice on my simulation strategy as well as feedback on my results, both of which played a key role in my project.

Thank you to the LEES staff and students for their support throughout the course of the project. In particular I would like to thank Minjie and Samantha for helping me run simulations when I was not physically in lab.

Finally, a special thanks to my family, friends, girlfriend, the Graduate Christian Fellowship, and God who all gave me encouragement and moral support during the project.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Introduction

Since its invention over a century ago, many improvements have been made to the internal combustion engine. While modern engines are far more powerful and fuel efficient than their predecessors, there are still inefficiencies in engine operation which can and should be addressed. As a result, new engine technologies are still a valid and important research topic.

One of the most promising technologies for improved engine performance is Variable Valve Timing (VVT). A conventional engine without VVT opens and closes its valves using a cam and a camshaft which is coupled to the crankshaft. A single valve opening and closing time relative to the crankshaft position is chosen for each valve; it is fixed regardless of engine speed. This fixed valve timing is an issue because the ideal valve opening and closing times are strongly dependent on engine load and speed. The result is that these engines have optimal valve timings for a single specific engine load and speed, but run sub-optimally for all others. With VVT, optimal valve opening and closing times can be achieved at every engine load and speed.

With the proposed benefits of VVT in mind, a novel electromechanical valve drive, the MIT EMV, was created by Woo Sok Chang [1] and further developed by Yihui Qiu [2] in the Laboratory for Electromagnetic and Electronic Systems (LEES) at the Massachusetts Institute of Technology (MIT). The MIT EMV actuator has been developed to produce variable valve timing in an engine, but the benefit of using it is yet unknown. The determination of this is the focus of this thesis.

1.2 Thesis Goals

The primary goal for this thesis is to determine quantitatively the benefit in engine performance that the MIT EMV provides over standard valve actuation. The performance metric chosen for this thesis is reduction in fuel consumption at part load operation. Computer aided simulation is used to determine the MIT EMV's effect on fuel consumption. To do so, the following objectives are proposed:

1. Determine a reference engine with which to evaluate the MIT EMV
2. Find a method with which to simulate engine performance
3. Model the reference engine and the MIT EMV within the engine simulation software
4. Determine a simulation strategy to best evaluate potential performance gains
5. Perform simulations and then review and discuss the results

1.3 Thesis Organization

The thesis is organized as follows:

Chapter 2 discusses background information on variable valve timing, including examples of various actuators and their reported benefits, followed by the motivation for this project. The reference engine model and the MIT EMV model are then discussed in Chapter 3. Next, Chapter 4 covers the method for simulation and simulation strategy that will be used to best evaluate the MIT EMV performance. The results of this simulation strategy as well as a brief discussion of those results are covered in Chapter 5. Chapter 6 evaluates the thesis goals and proposes future work with regard to the MIT EMV.

Chapter 2 Background

2.1 Introduction

This chapter discusses the details of Variable Valve Timing (VVT) and Variable Valve Actuation (VVA). Several different methods of VVA that have been implemented by major automotive manufacturers are reviewed. Emphasis is placed on how each actuator works and the improvement in engine performance each has demonstrated. The MIT EMV is reviewed and the motivation behind this thesis is given.

2.2 Variable Valve Timing

Variable Valve Timing has become a broad term for any type of modification to valve events. In fact, there are several different ways to vary valve timing in an engine. The valve event can be controlled to have variable phase, lift, or duration, as well as any combination of the three. Variable phase refers to the ability to adjust the opening and closing points of the valve event together without changing the duration. Variable lift refers to the ability to adjust the maximum displacement of the valve into the engine cylinder. Variable duration refers to the ability to adjust the closing event of the valve relative to the opening event. An illustration of each modification to the valve lift profile is shown in Fig. 2-1.

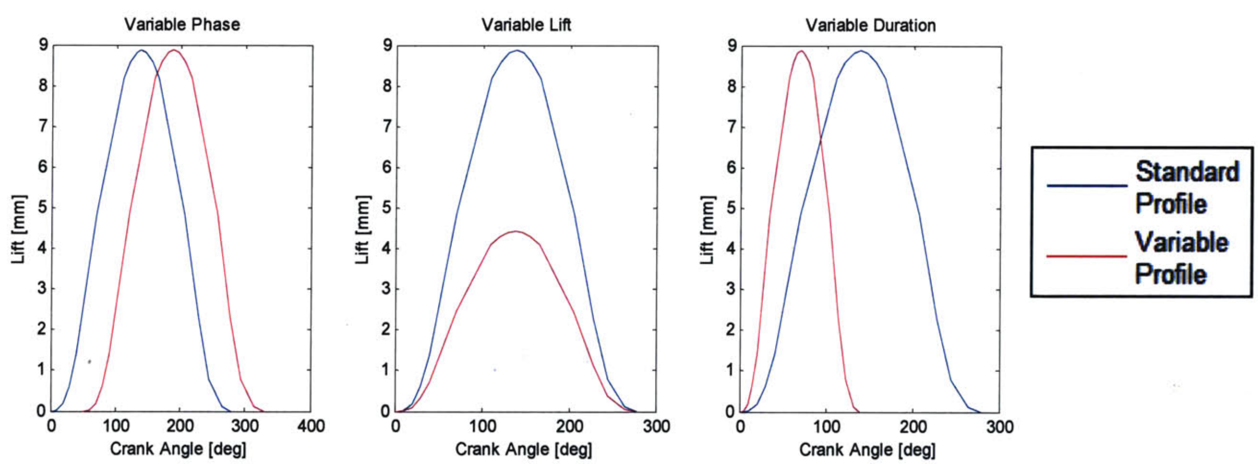

Fig. 2-1: Illustration of variable phase (left), variable lift (middle), and variable duration (right).

There are numerous benefits from having complete control over engine valves. The main benefit, and the focus of this thesis, comes from precise control over how much fresh air enters a cylinder in normal operation. At full load, the amount of air entering the cylinder is maximized by keeping the valve open as long as necessary. At part load, the valve timing can be used to limit the air in the cylinder, removing the need for a throttle plate and thus removing associated pumping inefficiencies. There are several potential benefits of VVT beyond the scope of this thesis, some of which include improved engine braking, exhaust residual control, and cylinder deactivation [**3**].

2.3 Methods for Evaluating VVT

Several methods have been used to determine what performance gains can be achieved with VVT and how exactly the valves need to be operated to do so. The most obvious method is to use an actuator on a physical engine. Using an electromagnetic actuator on a single cylinder research engine, Theobald and Lequesne were able to show that intake valve closing provides a gain in efficiency over traditional throttling [**4**]. In some cases, however, physical experimentation is not feasible, so computer aided simulation is used as a viable alternative.

An engine simulation code developed by Poulos and Heywood [**5**] was used by Assanis and Bolton to simulate the effect of valve timings on engine performance [**6**]. With the code, they were able to determine the optimum valve events for maximizing wide open throttle torque as well as experiment with valve timing strategies for load control without throttling. A gain of 11% in low speed torque and a fuel economy improvement of 9% at low load were found [**6**].

2.4 Evaluation of Variable Valve Actuators

While VVT is certainly beneficial, the best method for achieving it is still questionable. Many methods for variable valve actuation have been proposed. The focus here will be on actuators implemented by a few of the major automotive manufacturers, how they function, and the benefits gained. The actuators chosen do not represent the whole class of VVT designs, but were chosen as a subset which tries to incorporate the most commonly used actuators in the market.

One of the earlier forms of VVA is Honda's variable valve timing and lift electronic control system (VTEC) which employ's cam profile switching. The system uses a camshaft with two cams, one for low speed and another for high speed. A hydraulic piston is used to select which cam affects the valves. The technology allows for the engine to have high performance at high speeds without having to sacrifice low range performance including idling stability. Based on the data in [**7**], a Honda NSX engine with VTEC, when compared to similar conventional dual overhead cam (DOHC) engines, was able to achieve an estimated average of 25% increase in specific power output (kW/l) and an estimated average of 9% increase in specific torque output (Nm/L).

Another basic form of VVA is the use of variable camshaft phasers. Instead of directly coupling the crankshaft to the camshaft, an intermediate phaser is used which allows for intake and exhaust cams to be either advanced or retarded based on a hydraulic pressure control. Twin camshaft phasing was integrated into a 1.6L 4-cylinder 16-valve DOHC Ford engine, as shown in Fig. 2-2. With the twin phasing setup, an average torque increase of 10%, a maximum torque increase of 15%, and a maximum fuel economy improvement of 8 % were achieved when compared to a base engine [**8**].

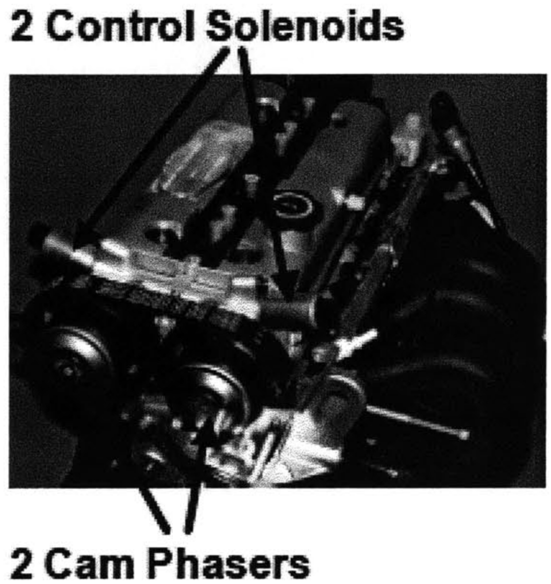

Fig. 2-2: Twin camshaft phasing applied to a 1.6L 4-cylinder 16-valve DOHC engine [**8**].

Combining the two previous technologies, Porsche's VarioCam Plus uses camshaft phasing along with valve profile switching. The lift profile is selected by the use of tappets which are controlled using oil pressure and an electro-hydraulic switch valve. The valve timing is adjusted via a geared camshaft adjuster which is also controlled with oil pressure and an axial plunger. Fig. 2-3 illustrates the lift and timing adjustment mechanisms. Testing the VarioCam Plus on a Porsche 911 Turbo Engine at different points on the engine map, 15% fuel economy gains in the emission-relevant range, 6-7 % less fuel use overall, and 35% higher full-load torque at 2000 rpm were achieved when compared to the previous Porsche 911 model without VarioCam Plus [**9**].

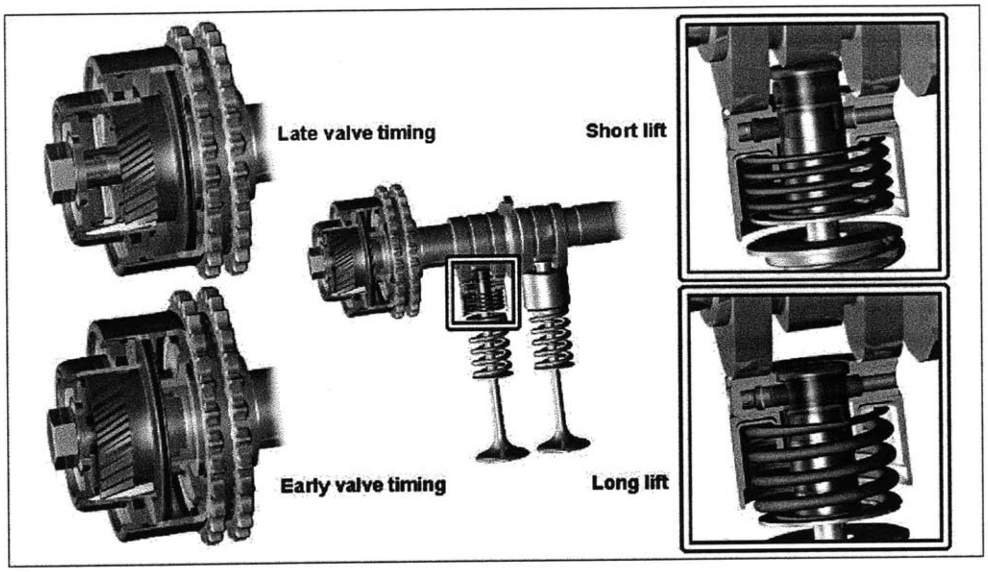

Fig. 2-3: Schematic of Porsche's VarioCam Plus mechanism [9].

Taking the combination seen in Porsche's VarioCam Plus one step further, BMW's Valvetronic system uses fully variable lift with cam phasing. Essentially, the valve lift is not constrained to two profiles but rather has infinitely adjustable inlet valve lift. This is accomplished by having the inlet camshaft act against an adjustable intermediate lever. The position of the intermediate lever defines the lift of the inlet valve. Because the intermediate lever's position is fully adjustable by a computer controlled eccentric shaft, the lift is therefore fully adjustable as well. Figure 2-4 shows the possible valve profiles; BMW's cam phasing mechanism, VANOS, can shift the profile from side to side while Valvetronic varies the lift of the intake valve. The improved intake valve lift control allows an engine with Valvetronic to run with throttle-less load control, the air coming into the cylinder is limited by the reduced valve lift. Using this strategy, Valvetronic can obtain a fuel savings of up to 20% at low loads and 10% overall [10].

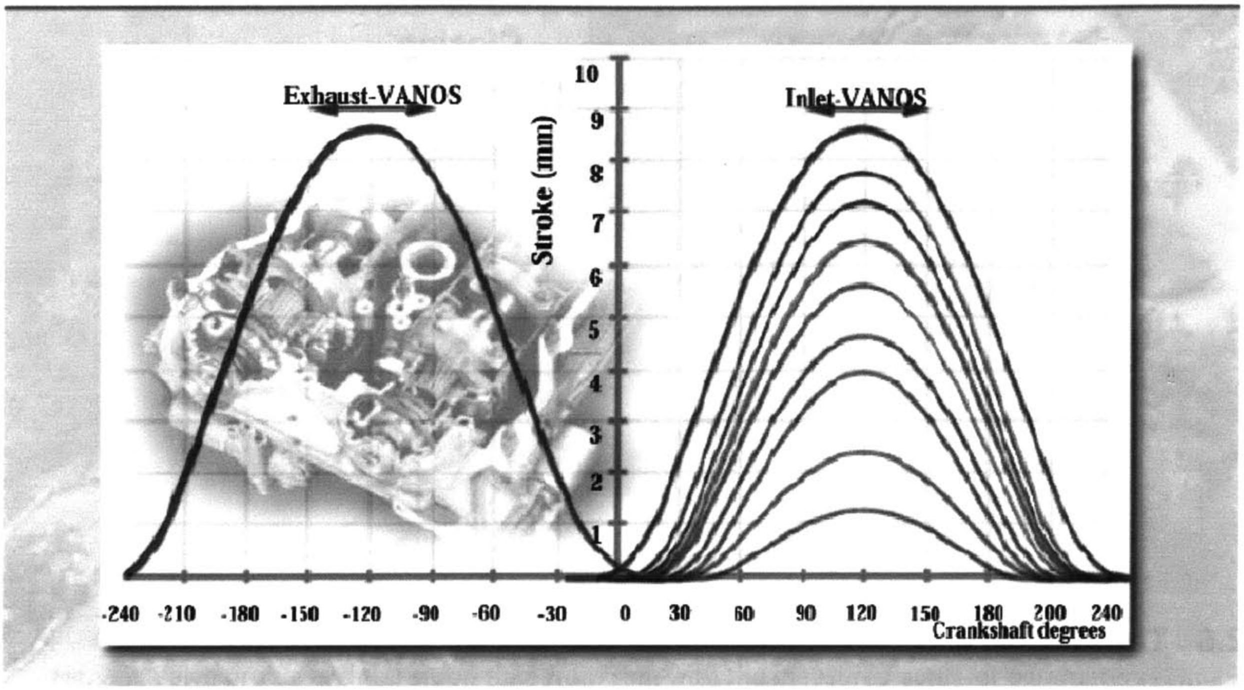

Fig. 2-4: Possible valve lift profiles achievable through BMW's Valvetronic technology [**10**]

The majority of variable valve actuators in production still use the traditional camshaft drive, but use mechanisms to adjust its effect on the valve timing and lift. An alternative to this is a standalone actuator which drives the valve directly such as the electromechanical valve drive (EMV) developed by FEV. The EMV uses electrical energy to drive a normal force actuator against regenerative springs to open or close the valve. Normal force actuators will increase in force as the valve approaches either extreme position, which is beneficial in that little power is required to hold the valve, but control of the valve speed is quite difficult. A schematic of the FEV EMV is shown in Fig. 2-5. Because the actuator is completely decoupled from the engine crankshaft, variable valve timing is inherent and throttleless load control can be accomplished by limiting the duration of the valve's open time. Testing the actuator in the New European Driving Cycle results in a 15% reduction in fuel consumption over a standard camshaft driven engine [**11**].

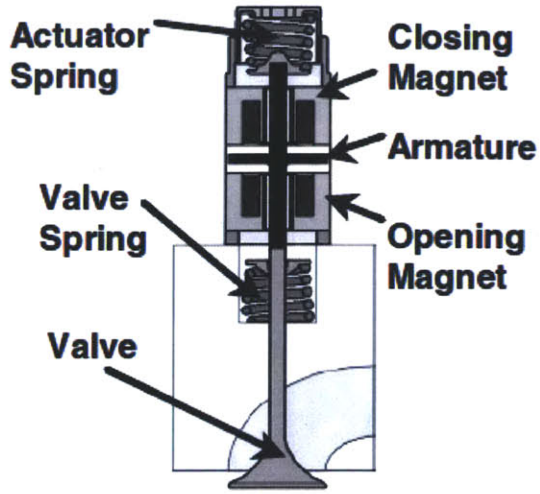

Fig. 2-5: Schematic of the FEV EMV [**12**].

2.5 The MIT EMV

The MIT EMV was developed to achieve features similar to the FEV EMV, but with more efficiency and a simpler control scheme by incorporating a shear force actuator. The MIT EMV uses electrical energy to drive a limited angle motor which acts against regenerative springs to move the valve. Fig. **2-6** shows how the MIT EMV uses a disk cam and roller to provide nonlinear motion of the valve. This valve lift profile, shown in Fig. 2-7, has two inherent benefits. First, when the valve is at either end of the profile, either fully open or closed, no necessary holding torque and therefore no power is required from the motor. Energy is only required when a transition occurs. Second, the profile was designed to give the valve an inherent soft landing when closing. The soft landing is important for avoiding noise problems and preventing wear on the valve [**1**].

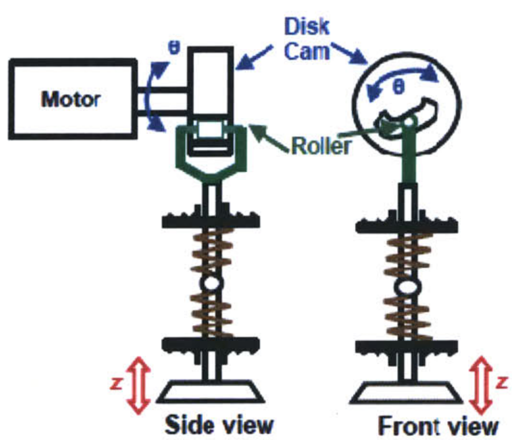

Fig. 2-6: Schematic of the MIT EMV **[2]**

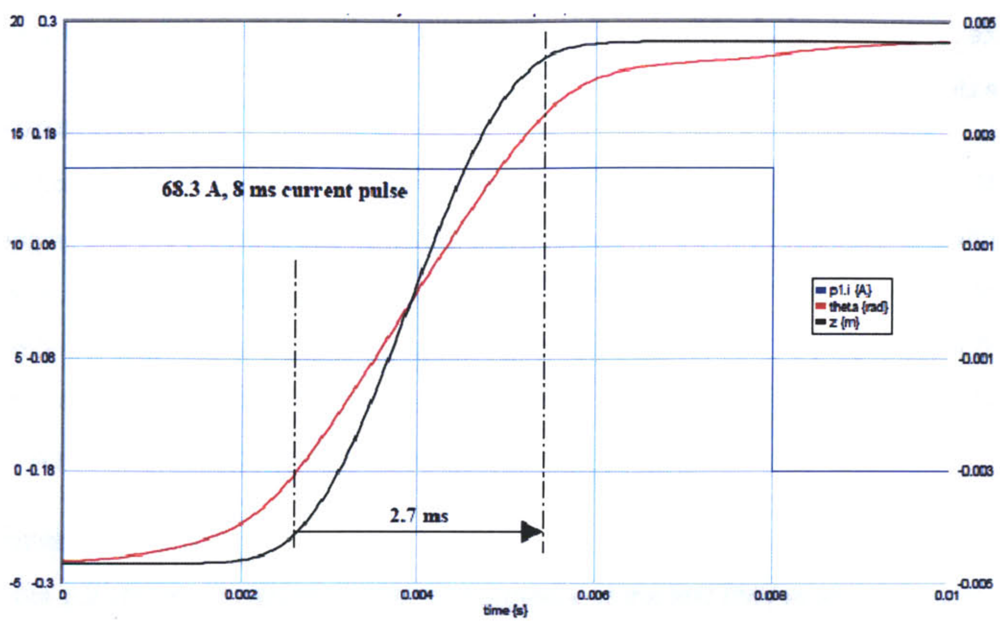

Fig. 2-7: Valve profile and transition time from sim. with the measured actuator parameters [**2**]. Theta and z are illustrated in Fig. **2-6** and p1 is the current pulse input into the motor.

Standard camshaft-driven valves have a transition time which is dependent on engine speed, but the MIT EMV motor was designed to achieve quick transition times of between 2.6 ms and 3.1 ms regardless of engine speed [**2**]. This transition speed is limited by the actuation speed of the motor and the stress on the valve. At high engine speeds, the two valve actuators will give similar transition times, but at lower speeds the MIT EMV will be much faster. This fast transition time at low speeds gives it more precise control to accomplish valve throttling. Similar to the FEV EMV, the MIT EMV can theoretically limit the air in the cylinder by controlling valve opening duration.

2.6 Motivation

The properties of the MIT EMV show great potential to improve engine performance, but the magnitude of improvement is yet unknown. At part load, fuel consumption can be reduced by minimizing throttling losses, and at full load, power output can be improved by providing better breathing. Based on other actuators, an improvement in the range of 10-20% in both fuel consumption and power output is to be expected, but results for the specific MIT EMV actuator are still necessary to prove its validity. The next step in the development of the MIT EMV is to run an engine simulation which incorporates the actuator. First the optimal valve timings and an optimal valve throttling scheme will have to be determined. Then, a simulation with those optimal timings will show just how beneficial the MIT EMV design is for engine performance . With the engine performance results, the practicality and value of the MIT EMV will be made more clear.

Chapter 3 Engine and Valve Train Models

3.1 Introduction

This chapter discusses the engine and valve train models used for simulation. First, the selection of reference engine and the choice of engine model that best works with the simulation software WAVE® is discussed. Next, the two valve train models are explained. The standard valve train model was taken from the reference engine. The MIT EMV model's lift profile was created from properties of the physical actuator with other flow and geometry properties taken from the reference valve model. Finally, the assumptions that went into the model are discussed.

3.2 Reference Engine

The goal of the simulation is to determine the performance of the MIT EMV relative to a conventional valve train. Two engine models are desired, one with a conventional valve train, and one with the MIT EMV. To best determine the benefits attributed to the MIT EMV, the same engine model is used as a base, with only the valve train parameters changed.

The choice of base engine is somewhat arbitrary due to the fact that the MIT EMV can be incorporated into just about any engine. When deciding on the type of engine to use, current practices in industry were considered. Automotive fleets have a wide range of engine sizes and generally the vehicles which are promoted as the most fuel efficient often have small 4 cylinder engines while those that are the most powerful have large 6 or 8 cylinder engines. Because the performance metric of the MIT EMV in this simulation is fuel consumption, it was decided that a small 4 cylinder model would be most useful.

The largest deciding factor for the engine model came from considerations based on the use of simulation software. The MIT Sloan Automotive Laboratory has made available the engine simulation package WAVE® by Ricardo. WAVE® is a market-leading simulation package used worldwide in industry sectors [**13**], which will help generate credibility and validity of any results obtained from it. Using WAVE®, however, introduces the need for an engine model specifically designed using the WaveBuild® software. The choices are then to create a model from scratch or to use a preexisting model. To create a model requires a strong knowledge of the WaveBuild® software as well as engine data to calibrate and verify the created model. To create such an engine model would require experience and time beyond the scope of this project, so using a preexisting model is the ideal choice. It is difficult to obtain an accurate and calibrated model from automotive companies because of the proprietary information contained in each model. Instead, a generic engine model which was included with the WAVE® software package was selected for use.

The generic model that was chosen from WAVE® is of a 1.57L, 4 cylinder, 16 valve SI PFI engine with maximum power of 103.1 kW BHP at 6000 rpm. This engine has the properties of a smaller, more fuel efficient engine and comes fully modeled in the WAVE® software. Based on geometry, temperature, and coefficients for heat transfer and friction, the intake, throttle, fuel injector, valves, piston, cylinder, exhaust, catalytic converter, and muffler are all modeled and can be seen in block diagram form in Fig. 3-1 [**14**]. The engine block contains additional combustion, friction, and heat transfer models.

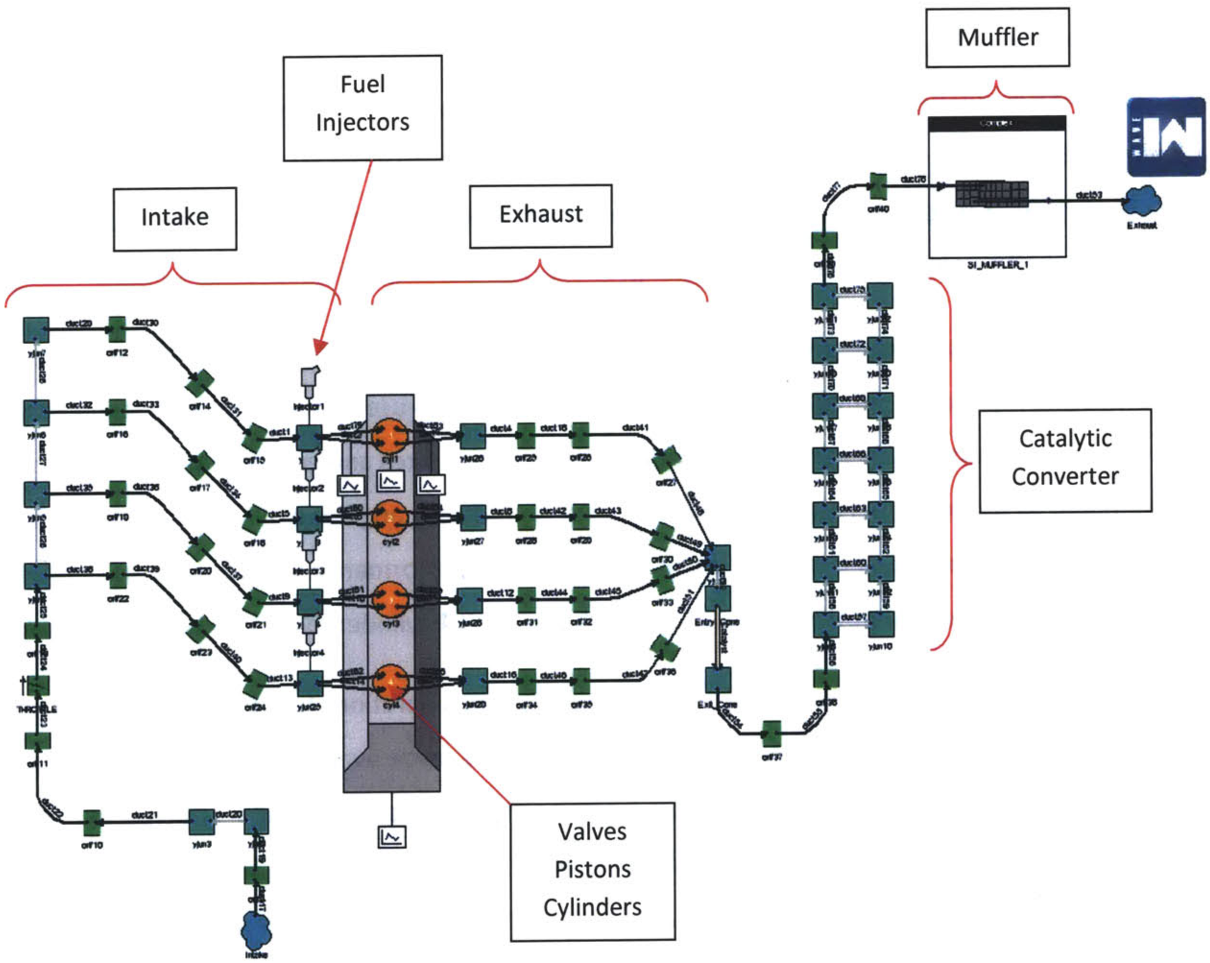

Fig. 3-1: WAVE® model showing various components and their corresponding block diagrams [14].

Combustion model

The primary combustion model in WAVE® is based on the SI Wiebe function [15] which gives the rate of fuel burned as a function of crank angle, stated in (3.1). Additionally, a 50% burn point is specified to move the curve relative to the TDC location. The reference engine model holds WEXP fixed at 2.0, but varies burn duration and the 50% burn point for each engine speed, the values of which are given in Table 4-1. A plot of a sample combustion profile is shown in Fig. 3-2. The combustion model assumes that fuel and air are fully mixed and will burn at a defined fuel/air ratio.

$$W = 1.0 - \exp\left(-AWI\left({}^{\theta}/_{BDUR}\right)^{(WEXP+1)}\right) \tag{3.1}$$

Where: *AWI* is an internal parameter to allow BDUR to cover the 10-90% range
θ is the crank angle past the start of combustion in degrees
BDUR is the 10-90% combustion duration in degrees
WEXP controls the curve to burn mass earlier or later

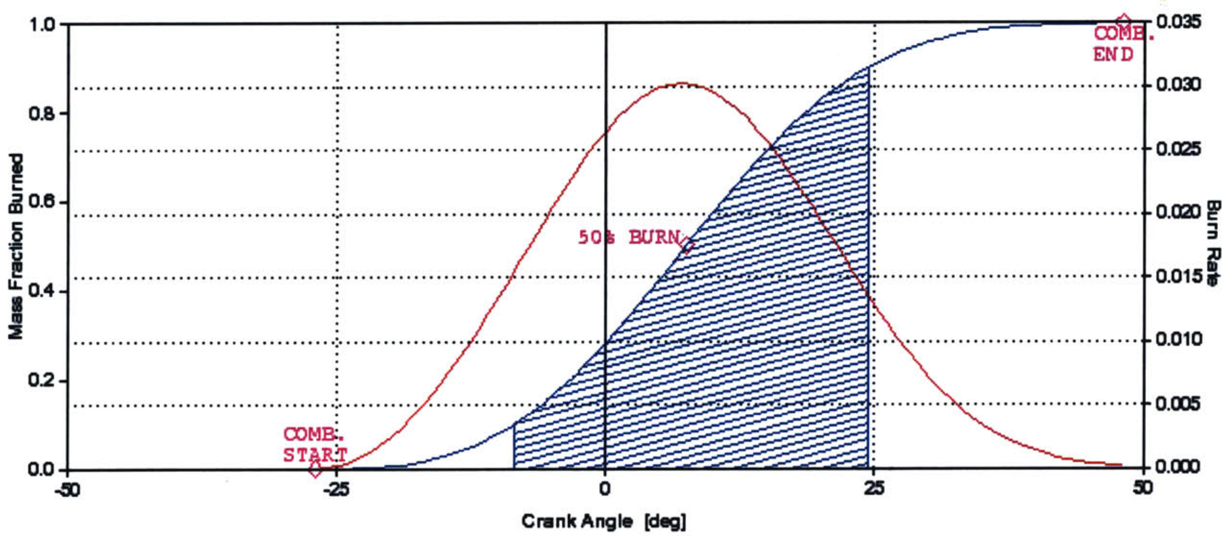

Fig. 3-2: Sample combustion profile showing the mass fraction burned (blue) and the burn rate (red).

Friction model

The friction model is based on a modified form of the Chen-Flynn correlation [16]. Accessory friction, peak pressure, hydrodynamic friction, and windage losses are taken into account as stated in (3.2). The reference engine model has constant correlation values for all engine speeds; A_{cf} is 0.3 bar, B_{cf} is 0.005, C_{cf} is 325 Pa.min/m, and Q_{cf} is 0.2 Pa.min^2/m^2.

$$FMEP = A_{cf} + \frac{1}{ncyl}\sum_{i=1}^{ncyl}\left[B_{cf}\left(P_{cyl}\right)_i + C_{cf} * \left(S_{fact}\right)_i + Q_{cf} * \left(S_{fact}\right)_i^2\right] \quad (3.2)$$

$$S_{fact} = RPM * stroke/2$$

Where:

A_{cf} is the constant friction correlation

B_{cf} is the peak cylinder pressure correlation

P_{cyl} is the maximum cylinder pressure

RPM is the cycle average engine speed

$stroke$ is the cylinder stroke

C_{cf} is the piston speed correlation to account for hydrodynamic friction

Q_{cf} is the piston speed correlation to account for windage losses

Heat transfer model

The heat transfer model is based on the original Woschni heat transfer model **[17]**. The model uses specified cylinder and valve temperatures to determine the heat transferred to and from the charge, assuming uniform heat flow coefficient and velocity on all surfaces of the cylinder. The Woschni heat transfer coefficient is defined by (3.3). For the reference engine model, the variable scaling factor , C_{enht}, is only used to adjust the cylinder head surface by scaling it by 1.6.

$$h_g = 0.0128D^{-0.20}P^{0.80}T^{-0.53}v_c^{0.8}C_{enht} \quad (3.3)$$

Where:

D is the cylinder bore

P is the cylinder pressure

T is the cylinder temperature

v_c is the characteristic velocity of the charge

C_{enht} is a variable scaling factor

3.3 Reference Valve Model

Among the several engine parameters included with the model is a set of standard valve timings. There is one option to have constant valve timing and another for cam switching for the intake valve. The latter case switches profiles at speeds above 5000 rpm. Because the simulations for this project will all occur in the low 700-3000 rpm range (discussed in Chapter 4), the cam switching option is not necessary. Constant valve timing options taken from the reference engine are used for both the intake and exhaust valves. The intake opens at 330° CA with a duration of 280° CA and a lift of 12.57 mm (0° CA is TDC after compression). The exhaust valve opens at 105° CA with a duration of 300° CA and a lift of 8.64 mm. The lift profile is defined by tabular values but conforms to the standard bell shaped curve created by a standard cam profile. Figure 3-3 shows the entire intake and exhaust lift profiles for the standard valve train that is used by the reference engine model.

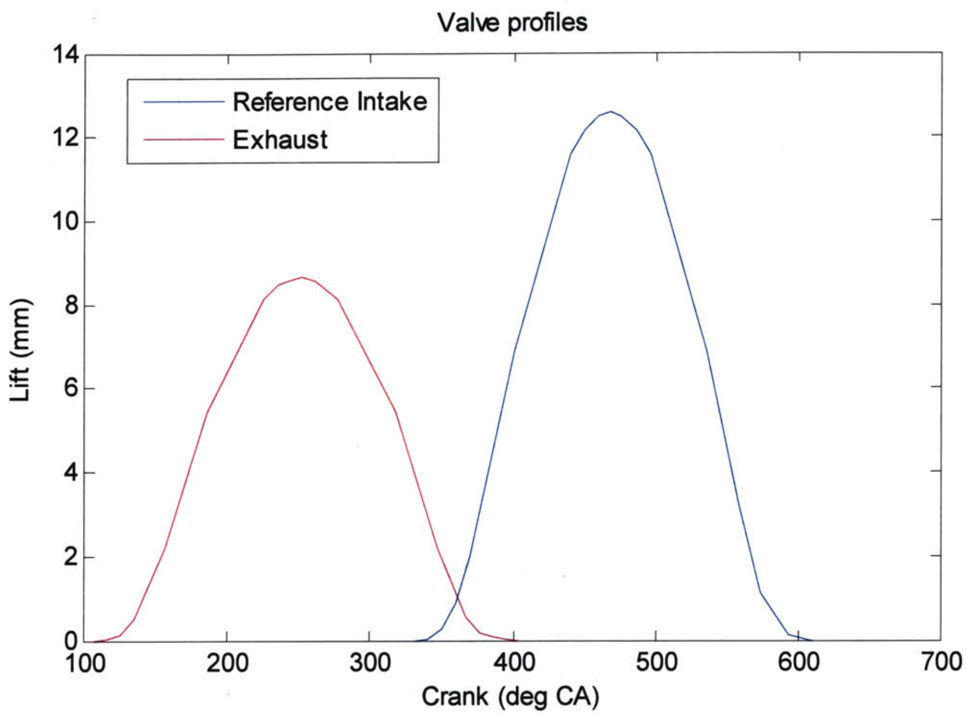

Fig. 3-3: Reference intake and exhaust valve lift profiles.

3.4 MIT EMV Model

The WAVE® simulation package models valves using their diameter, lift, flow, and swirl properties. To incorporate the MIT EMV into the reference engine, the valve lift profile from the reference engine was replaced with the valve profile of the MIT EMV. The new profile is no longer fixed, so a different profile is necessary for each desired set of valve timings and engine speeds.

Whenever possible, the properties of the MIT EMV were taken from the design, simulation, and physical testing of the actuator. The valve drive was original developed by Dr. Woo Sok Chang [1] and was later improved by Dr. Yihui Qiu [2]. The main parameters which were taken from their work are the valve's path of travel and the amount of time that it takes the valve to travel the path.

Lift Path

The EMV valve's path is defined by the cam in the actuator. The original cam was designed to have the valve move as the motor rotates between a range of ±26° with a smooth transition at either end. The equation defining the original valve's lift path relative to motor angle is given in (3.4).

$$z = f(\theta) = \begin{cases} 4\sin(3.46\theta)\ \mathrm{mm} \\ 4\,\mathrm{sign}(\theta)\ \mathrm{mm} \end{cases} \text{if} \begin{array}{l} |\theta| \leq 26^\circ \text{ or } 0.454\mathrm{rad} \\ |\theta| \geq 26^\circ \text{ or } 0.454\mathrm{rad} \end{array} \tag{3.4}$$

The work of Dr. Qiu included improvement of the cam and valve lift path. The ±26° rotation was reduced to a ±15°rotation which results in a faster transition time and a decrease in power of about 40% [2]. The new lift path as a function of motor angle is given in (3.5). A comparison of the two paths is shown in Fig. 3-4.

$$z = g(\theta) = 4\frac{\sin(6\theta)}{\sin(0.999\pi/2)}\ \mathrm{mm} \quad |\theta| \leq 0.262\mathrm{rad}\ (15^\circ) \tag{3.5}$$

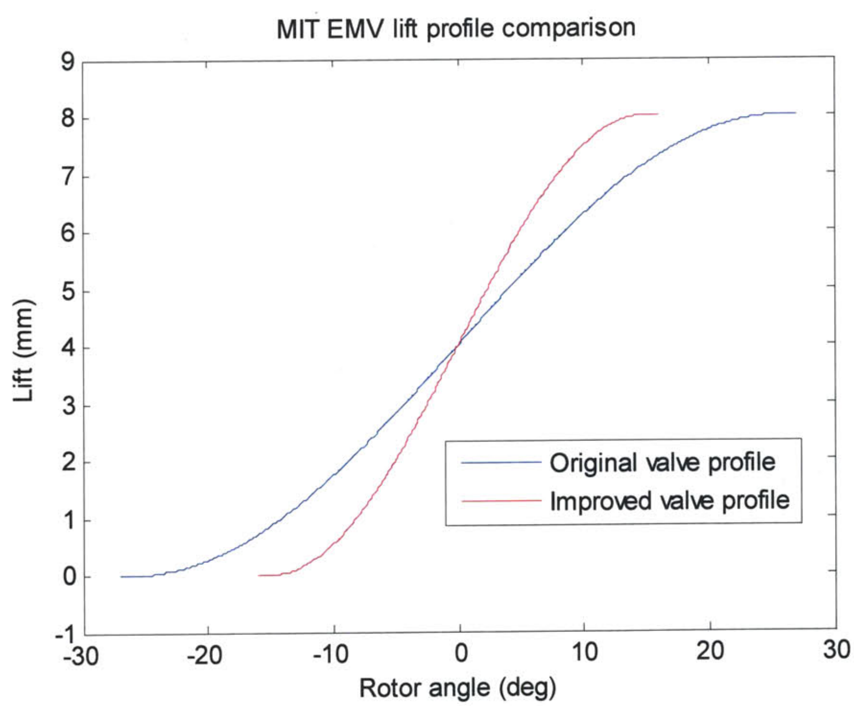

Fig. 3-4: Comparison of the original and optimized valve profiles for the MIT EMV

Transition Time

The other important feature in creating the valve lift profile is the time of each valve transition, the time it takes for the valve to travel from 5% to 95% of the lift. Unlike the traditional camshaft actuated valve drive, the MIT EMV transition time is not dependant on engine speed, but is fixed. The first valve drive design by Dr. Chang was able to achieve a transition time of 3.5 ms which was determined to meet the desired 6000 rpm operation transition speed requirement [**1**]. As mentioned, the improvements made by Dr. Qiu included an inherent faster transition time, which gives more flexibility in valve operation. The new design was evaluated under different configurations and gas forces, and the transition time for an intake vale with negligible gas force was determined to be 2.7 ms [**2**]. Even though these fast transition were designed for 6000 rpm operation, they are just as relevant for low speed operation because the transition time is exactly the same at every engine speed.

Lift Profile

With lift path and transition time data, the valve lift profile for the MIT EMV can be obtained for just about any desired valve opening and duration combination. The profile is constructed by first traveling the lift path, then staying at a constant lift where the valve remains open, and then following the lift path back down. The crank angle that it takes for the valve to move from fully closed to fully open (and vice versa) is defined by the valve transition time and engine speed as stated in equation (3.6).

$$°CA = (t\,[\mathrm{ms}]) \cdot \left(\frac{1}{60{,}000}\left[\frac{\mathrm{min}}{\mathrm{ms}}\right]\right) \cdot \left(N\left[\frac{\mathrm{rev}}{\mathrm{min}}\right]\right) \cdot \left(360\left[\frac{°\mathrm{CA}}{\mathrm{rev}}\right]\right) = 0.006 \cdot t \cdot N \tag{3.6}$$

Where *t* is the transition time in ms and *N* is the engine speed in rpm.

The amount of time that the valve remains in the fully open position is defined by the remaining crank angle that is required to meet the desired valve duration. An example MIT EMV lift profile is given in Fig. 3-5 along with a valve profile from the reference valve for comparison, both are for the same opening and duration. While not obvious from the figure, the MIT EMV lift profile does have smooth, flat ends to ensure soft landing.

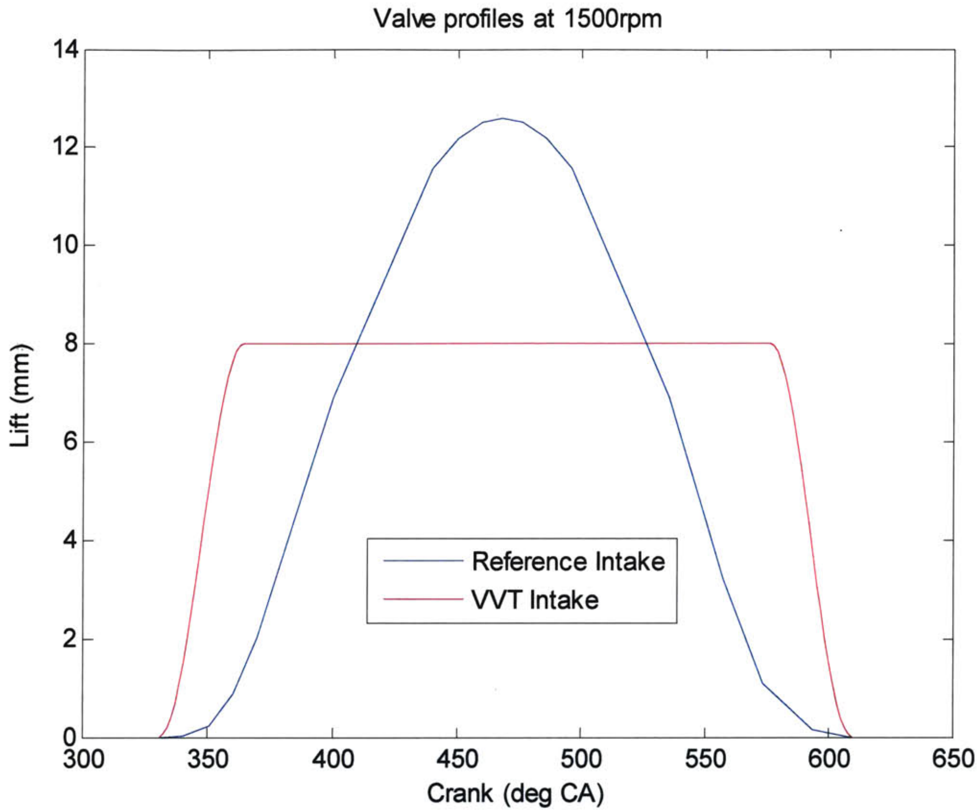

Fig. 3-5: The reference valve and the MIT EMV's lift profile at the same opening and closing points.

The above valve profile construction process only works when the total valve duration time is greater than the amount of time it takes the valve to move from closed to open and back to closed, i.e. one full valve event. While a single valve transition period is only 2.7 ms, back-to-back transitions require a total of 8.3 ms to travel from 5% to 100% and then back to 5% of the total lift [**2**]. The additional time for the turnaround duration is due to the flat ends designed into the MIT EMV's nonlinear cam profile [**2**]. If a faster back-to-back transition is necessary (especially at higher engine speeds), Dr. Qiu proposed a method of reducing turnaround time by starting the closing processes before the valve is fully opened, resulting in a reduced maximum lift. The 8.3 ms back-to-back transition can be reduced to 6.8 ms if the valve travels from 5% to only 98% and back to 5% of its possible lift. Simulation plots of these two types of back-to-back transitions are shown in Figs. 3-6 and 3-7.

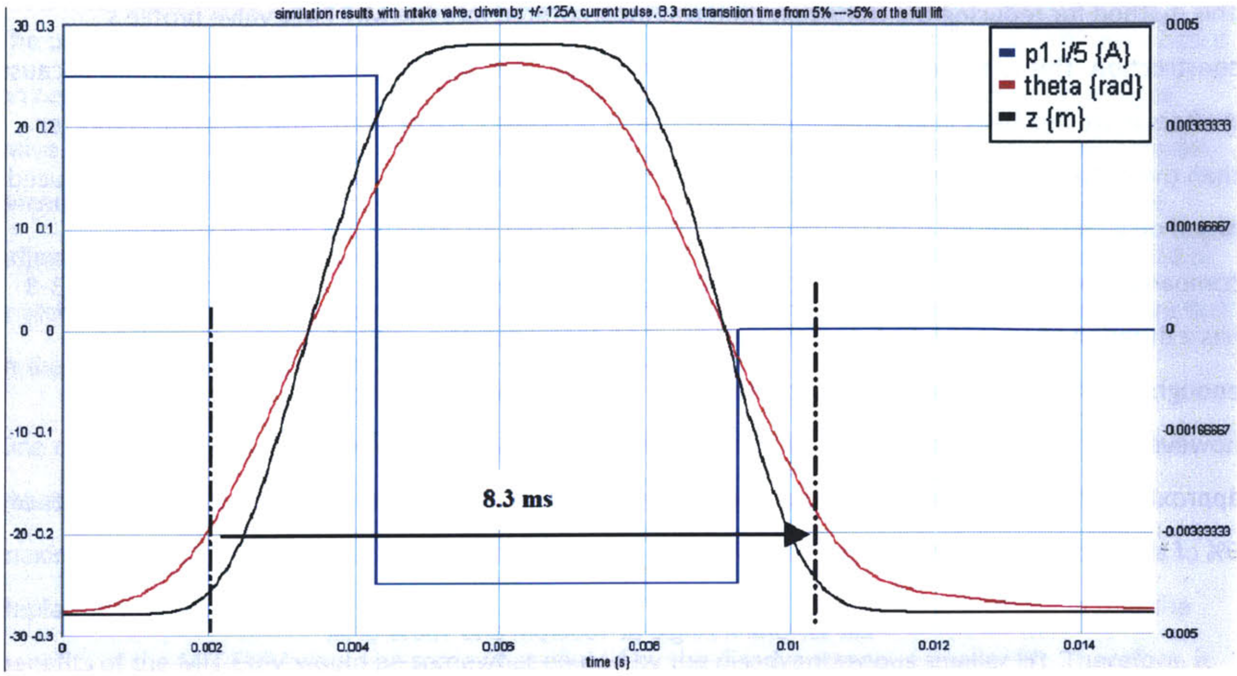

Fig. 3-6: Back-to-back transition in 8.3 ms with valve traveling to 100% of its lift.

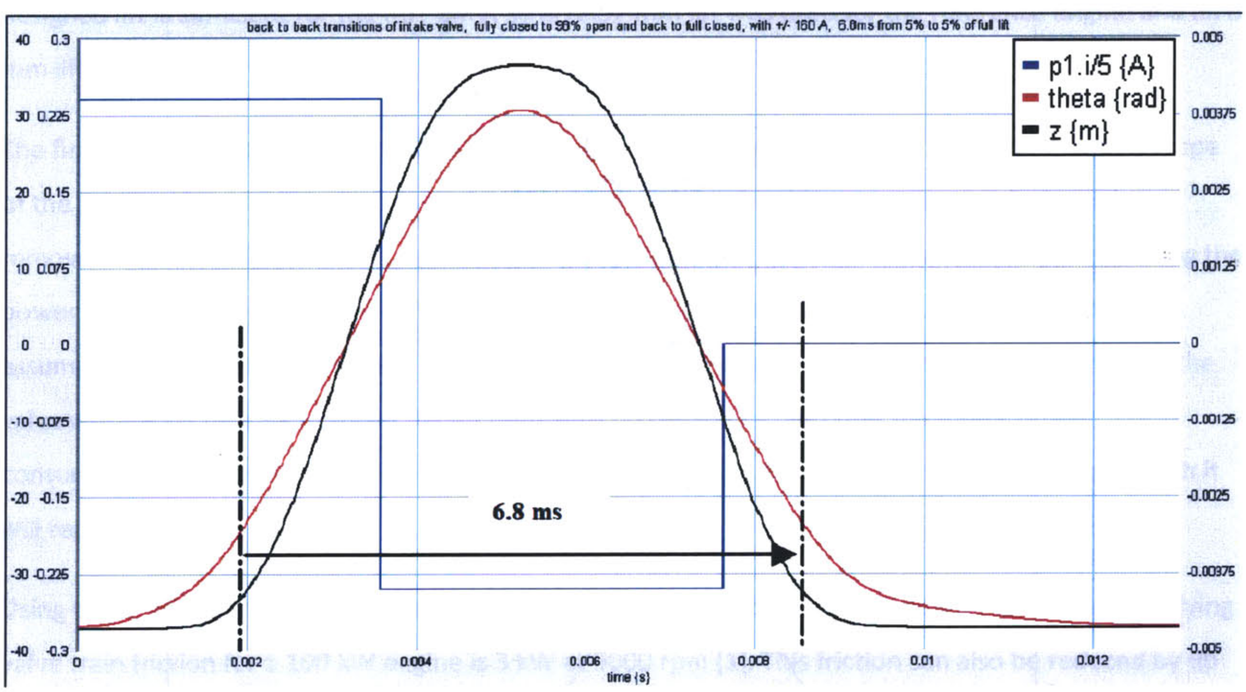

Fig. 3-7: Back-to-back transition in 6.8 ms with valve traveling to only 98% of its full lift.

This method for reducing the transition time was incorporated into the MIT EMV valve profile's construction. This reduced time is necessary for high speed, but at low speed it is still important because it offers more flexibility in allowing even shorter valve durations. If a desired valve duration is shorter than the crank angle equivalent of 8.3 ms as determined through equation (3.6), then the lift is reduced based on a linear interpolation between 98% and 100% of the lift. Figures 3-8 and 3-9 show a comparison between the MIT EMV model's full lift and reduced lift scenarios. The lift profile in Fig. 3-8 has a duration of 100°CA which corresponds to an 11.11 ms duration at 1500 rpm, so the valve has enough time to perform a back-to-back transition without reducing the lift. The profile in Fig. 3-9, however, has a duration of 62°CA which corresponds to only 6.89 ms, so the lift is reduced to approximately 98% of the full lift. Note that the times here correspond to starting and ending points of 0% of the lift rather than 5%.

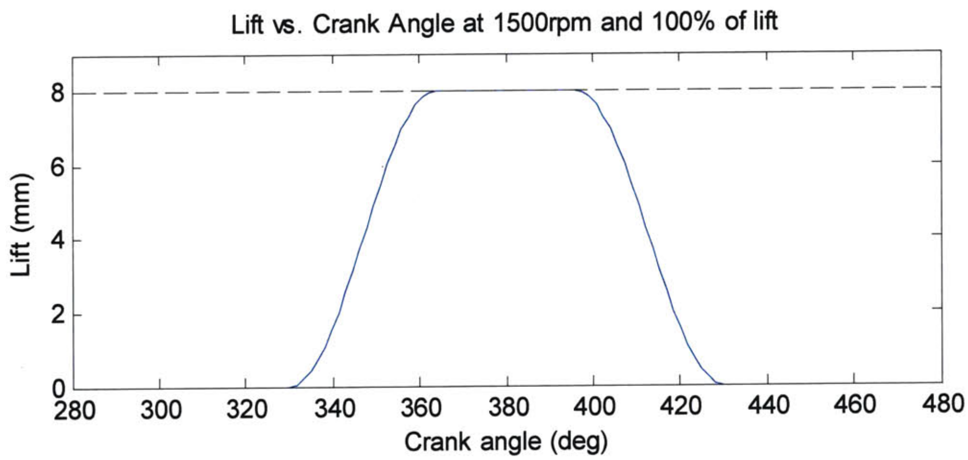

Fig. 3-8: Valve profile with a duration of 100°CA and reaches the full 100% of the lift.

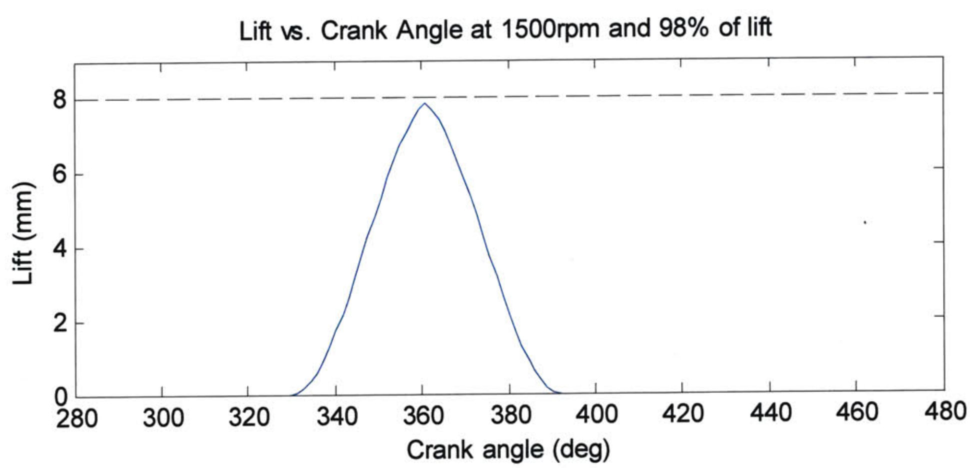

Fig. 3-9: Valve profile with a duration of only 62°CA and reaches only 98% of the full lift.

Model Assumptions

The portions of the model that were not taken from previous development of the MIT EMV were based on certain assumptions which either simplify the problem or match characteristics of the reference valve. In addition to the lift profile, geometry and flow properties are needed to specify a valve model. Because the MIT EMV is a unique valve drive, the choice of actual valve that is being driven can be adjusted. Therefore, valve geometry such as diameter and flow profile over the valve were selected to match that of the reference valve. The assumption is that the reference valve would be able to function in the MIT EMV with no penalty.

One concern about the MIT EMV is the maximum lift of the valve. The 8 mm lift is designed into the valve drive through the nonlinear cam. The reference engine, however, uses a lift profile with a maximum of 8.89 mm which is then scaled by a factor of 1.414 resulting in a total lift of 12.57 mm. Replacing that large lift valve model with the MIT EMV's lift model could cause a problem in that the benefits of the MIT EMV would be somewhat diluted by the disadvantageous smaller lift. Therefore, it was determined to run tests with the two lifts made equal. The standard 8 mm lift profile for the MIT EMV was then scaled by a factor of 1.571 at every point. The results of using the two valve lifts, 8 mm and 12.57 mm, for the MIT EMV model is shown and discussed in Chapter 5. It was determined that the designed lift is sufficient for the MIT EMV, so a 12.57 mm lift was used for the reference engine and an 8 mm lift was used for the MIT EMV.

The final considerations come from the power requirements of the MIT EMV. Due to the limited scope of the project, the MIT EMV is only implemented with the intake valves, so one camshaft will be removed and one will remain for the exhaust valves. Significant work has already been done to bring the power consumption of the MIT EMV down to levels comparable to traditional camshafts. One assumption is that the intake valve will operate without any opposing gas force, which means that the valve will always be able to travel the lift path with no resistance, and that the motor power consumption can be neglected if it is comparable to the friction force of a standard valve train which it will replace.

Using the breakdown of available energy in an engine as described in [18], the average power supplying valve train friction for a 100 kW engine is 3 kW at 6000 rpm [1]. This friction can also be reduced by up to 50% by roller cams [1], so the camshaft friction requirement of just the intake is estimated to be 750 W. Simulation results for the MIT EMV opening and closing an intake valve without gas force show an average power consumption of 52 W per valve over one cycle at 6000 rpm [2]. Therefore, driving the eight valves in the engine with the MIT EMV would require 416 W of power. At high speeds, the MIT EMV will actually save on power but only by approximately 0.33% of the 100 kW of brake power. An assumption is then made that the difference in power consumption of the two valve drives is negligible at all engine speeds, and the standard friction model in WAVE® will be used to calculate the power requirements for both models.

Chapter 4 Simulation

4.1 Introduction

This chapter discusses the method for simulating the MIT EMV using the engine and valve train models. First, the selection process of the engine simulation software is covered. Next, interfacing with MATLAB® including methods for maintaining model parameters and assumptions is explained. Then, details are given for the simulation strategies of varying valve timing and duration. Finally, a summary is given of the different simulations that were run throughout the course of the experiment, the results of which are discussed in Chapter 5.

4.2 Engine Simulation Software

Much of engine and engine component evaluation requires experimental results, and quantifying the benefits of the MIT EMV is no different. A physical experiment with the MIT EMV mounted directly on an engine head would be ideal, but unfortunately this was not an option due to the development and time needed to do so. While some theoretical models exists for the heat transfer, friction, fluid flow, and the like within an engine, a combination of theoretical and empirical models is required to best describe the complexities of everything that goes on in an engine. Therefore, it was determined to use a preexisting computer simulation package which incorporates validated engine operation models.

Engine Simulation Code

The first consideration was a spark ignition engine cycle simulation code developed by Poulos and Heywood [5], which was also used in 1994 by Assanis and Bolton to simulate the effect of valve timings on engine performance [6]. This code was offered and is often used by the Sloan Automotive Laboratory at MIT. For a series of inputs such as engine speed, air to fuel ratio, manifold pressure, etc., the code will output the amount of fuel consumed, residual gas remained, heat transferred, and work produced over an engine cycle. An interface was developed to use this program with MATLAB® and some test results were generated, but ultimately this code was not used. While this simulation method should be sufficient for determining some benefit from the MIT EMV, the credibility of those results among researchers outside of MIT is questionable, and more advanced engine simulation packages are available.

Ricardo WAVE®

The Sloan Automotive Laboratory at MIT has licenses for two major engine simulation packages, WAVE® by Ricardo and GT-Power® by GTI. For the purposes of simulating the MIT EMV performance, either program would suffice and WAVE® was chosen. WAVE® has several characteristics which set it apart from a generic engine simulation code. In addition to advanced engine combustion modeling, WAVE® provides detailed models for intake and exhaust pathways, valve profiles, and fluid flows. The most prominent factor in choosing WAVE® is that it has been accepted and used by industry and comes with that additional credibility.

4.3 Interfacing with MATLAB®

While the modeling and simulation for this experiment was done in WAVE®, the pre- and post-processing for each test was done in MATLAB®. This method came from the initial use of the Poulos and Heywood code for which the code was run as a Fortran executable which reads an input text file and writes an output text file. MATLAB® code was written to generate a desired input text file, run the executable, and then read the output file and import the data into MATLAB® for further use. When the decision to use WAVE® was made, this infrastructure was again implemented. While WAVE® has a series of tools for working with data, including its WavePost® program, MATLAB® works very well for manipulating data with greater control and familiarity. The following is a brief description of the input and output workflow using MATLAB® with WAVE® for any type of simulation, either full load with better breathing or part load with valve throttling.

Input

The necessary input parameters for each WAVE® simulation consist of valve profile, combustion, geometry, and temperature properties. Apart from the valve profile data, the specified properties are obtained from the reference engine model. While most of the reference engine parameters are fixed, some vary with engine speed and the model includes data for these parameters at eight different speeds. Table 4-1 gives this speed dependant data. Note, some properties are included in the table as variables but actually remain constant. When inputting parameters for a specific simulation, the data for each property was obtained from this table, using interpolation or extrapolation based on engine speed when required.

Table 4-1: Input parameters.
In descending order: Engine Speed, Air/ Fuel Ratio, Combustion Duration, Location of 50% Burn Point, Catalytic Converter Wall Temperature, Exhaust Valve Temperature, Cylinder Head Temperature, Intake Valve Temperature, Cylinder Liner Temperature, Piston Top Temperature, Runner Length, and Throttle Angle.

SPEED[rpm]	8000	7000	6000	5000	4000	3000	2000	1000
A_F	14.7	14.7	14.7	14.7	14.7	14.7	14.7	14.7
BDUR[deg]	33	32	31	32	31	29.5	29	28
CA50[deg]	7.5	7.5	8	8.5	8.5	9	9	9
CAT_TEMP[K]	1135	1120	1075	1025	950	900	830	790
EV_TEMP[K]	388	383	380	377	375	370	366	365
HEAD_TEMP[K]	648	639	635	630	620	595	580	550
IV_TEMP[K]	325	322	320	320	320	318	316	312
LINER_TEMP[K]	628	622	615	600	595	580	570	540
PISTON_TEMP[K]	609	603	595	585	580	550	530	500
RUNNER_LENGTH[mm]	188	188	188	188	188	188	188	188
THROTTLE_ANGLE[deg]	90	90	90	90	90	90	90	90

Engine load is not included in Table 4-1 because the power produced by the engine is a function of the speed and throttle angle. For the MIT EMV simulation, the power will be a function of the speed and valve timing since the engine will be operating at wide-open throttle. Therefore, to "input" a specific engine load, a corresponding valve timing would have to be specified.

In addition to the twelve variables in Table 4-1, an additional 88 variables are used to specify the crank angle vs. lift data points of the valve profile. The valve profile discussed in Chapter 3 is generated and inputted as a discrete set of crank angle vs. lift points. Along with satisfying the constraints of the model, extra criteria were implemented so that an arbitrarily created profile does not cause failure in the engine or the MIT EMV. First, any valve opening and duration combination cannot result in the valve being open when spark is initiated. This is satisfied by checking that the final crank angle of the generated lift profile is less than the burn starting crank angle. A conservative burn starting angle is calculated in (4.1) by subtracting the entire burn duration from the half way point as well as removing an extra 5°.

$$CA_{end} < CA_{50} - BDUR + 720° - 5° \quad (4.1)$$

Where CA_{end} is the last crank angle of the valve lift profile
CA_{50} is the 50% burn point
and $BDUR$ is 10% to 90% burn duration

The second criterion restricts the amount of valve lift reduction. As mentioned in Chapter 3, the minimum valve duration can be reduced if the valve does not open fully before beginning to close. This reduction was designed to accommodate high engine speeds, but is also important for determining minimum valve duration at any speed, so therefore, the criteria is always implemented. Theoretically, any valve duration would be possible as long as it is coupled with reduced lift, but in practice, the MIT EMV is not capable of such behavior. Because the actuator has been shown only to accomplish partial variable lift of 2% at minimum duration, only a 98% or greater lift profile is used in simulation. When running through a range of valve openings and durations, any profiles that do not meet these two criteria are ignored and left out of the simulation.

Once all of the necessary variables are calculated for a specific engine speed and valve timing, they are put into a table as a specific "case". To vary valve profiles over a series of opening and closing points, a couple thousand cases would be generated, copied into WAVE®, executed, written to an output file, and then imported into MATLAB® for post processing.

Output

After running a simulation, WAVE® produces a summary file which contains thousands of engine parameters based on the inputs for each case that was run. MATLAB® is used to read this summary file, search for relevant engine parameters, and import them as working variables for further manipulation.

For the part load simulations, the engine parameters for each case are imported in MATLAB® , and the optimal case and corresponding valve events are found for a desired load. First, each case is filtered to meet load and residual criteria, both of which are outputs of the WAVE® simulation. Because the load is not controlled with a throttle plate but rather with the valve timing, each case will generate a different amount of engine power. The Brake Mean Effective Pressure (BMEP) for each case is filtered to be within 0.5% of the desired BMEP. Also, the residual gas from each case must be less than 25%; anything higher than about 30% might still be acceptable with WAVE® but would affect combustion stability in an actual engine [18]. After these two criteria are satisfied, the remaining cases are then compared and the one with the lowest Brake Specific Fuel Consumption (BSFC) is selected as optimal.

Summary

With this methodology, the minimum BSFC for any engine operating point, defined by a given engine speed and load, can easily be found. Using the desired engine speed, a large range of valve opening and closing combinations can be generated and then simulated in WAVE®. The power output and residual gas constraint for each of those valve combinations are then analyzed to ensure that the desired load is satisfied. Finally, an optimal choice of valve opening and closing is made based on which has the lowest resulting BSFC.

4.4 Simulation Strategies

Ideally, optimal fuel consumption would be determined at every engine speed by checking every possible combination of valve opening and closing. Unfortunately, the required computing time to perform this task prevents it from being feasible. Therefore, a strategy for determining the benefit of the engine outfitted with the MIT EMV over the reference engine is necessary. This section describes how that strategy was selected and implemented.

Operating points

One possible metric for evaluation is to compare the two engines over an FTP (Federal Test Procedure) drive cycle. This test mimics actual vehicle operation by driving the vehicle over a specified speed range as shown in Fig. 4-1 [**19**]. The difficulty with using this test is that it defines vehicle speed over time rather than engine speed over a range of loads. Therefore, using the FTP cycle would require some vehicle modeling including assumptions for vehicle weight, transmission, drivetrain, etc. This additional complexity would introduce a large amount of uncertainty in the accuracy of the model. It was instead decided to use a set of key operating points which would best represent the cycle.

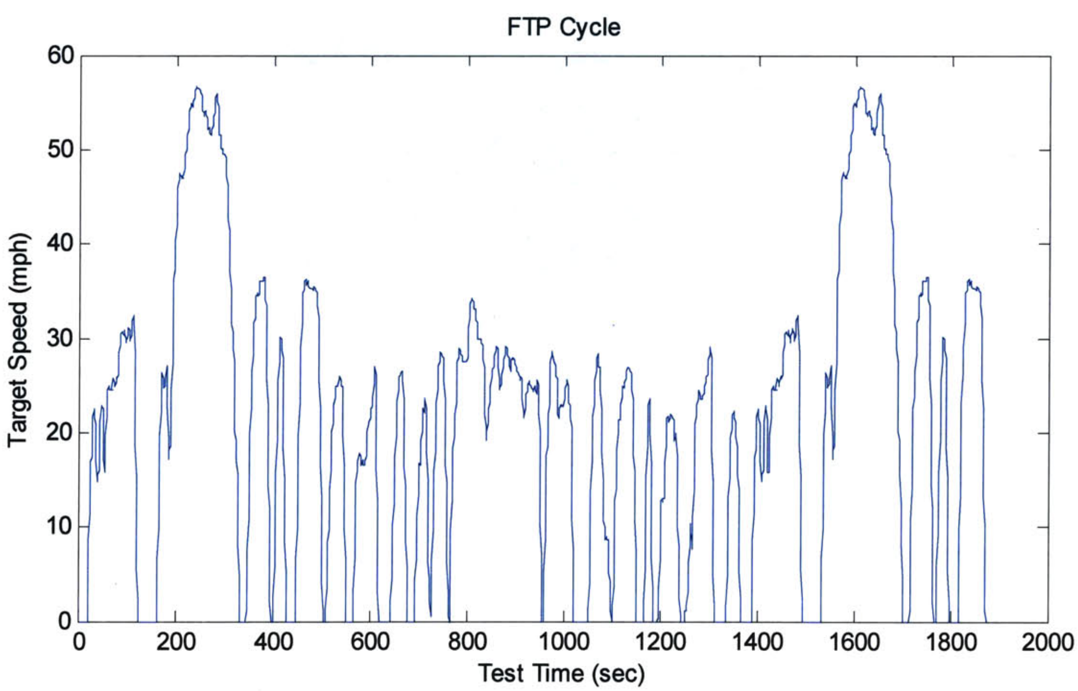

Fig. 4-1: FTP driving cycle [**19**].

Most passenger vehicle engines operate below 1800 rpm and 6 bar BMEP on the FTP test cycle [**20**]. Therefore, selecting operating points in this region would allow for a good approximation of operating over the cycle. Several automotive companies have corporate mapping points that are used to evaluate an engine: Chrysler uses an operating point of 1600 rpm and 2.41 bar BMEP, Ford uses 1500 rpm and 2.62 bar BMEP, and GM uses 1300 rpm and 2.95 bar BMEP [**20**]. In addition, an old standard operating point of 1500 rpm and 2.5 bar BMEP is often used in industry [**21**]. Optimizing the BSFC and valve events around these four operating points is a great opportunity to relate the benefits of the MIT EMV to engine manufacturers.

While these operating points work well for relating to engine manufacturers, it is also important to relate the benefits of the MIT EMV to other valve drives on the market. BMW's Valvetronic VVT technology has been incorporated into an engine and its benefits in fuel consumption are shown on the engine map in Fig. 4-2. Three distinct operating points and the respective fuel consumption benefits can be chosen from the map: at 700 rpm and 0.5 bar there is an increase of 18%, at 1700 rpm and 2.5 bar there is an increase of 10%, and at 2800 rpm and 3.5 bar there is an increase of 6%. Therefore, it was determined to also use these same three operating points with the MIT EMV to see how it compares.

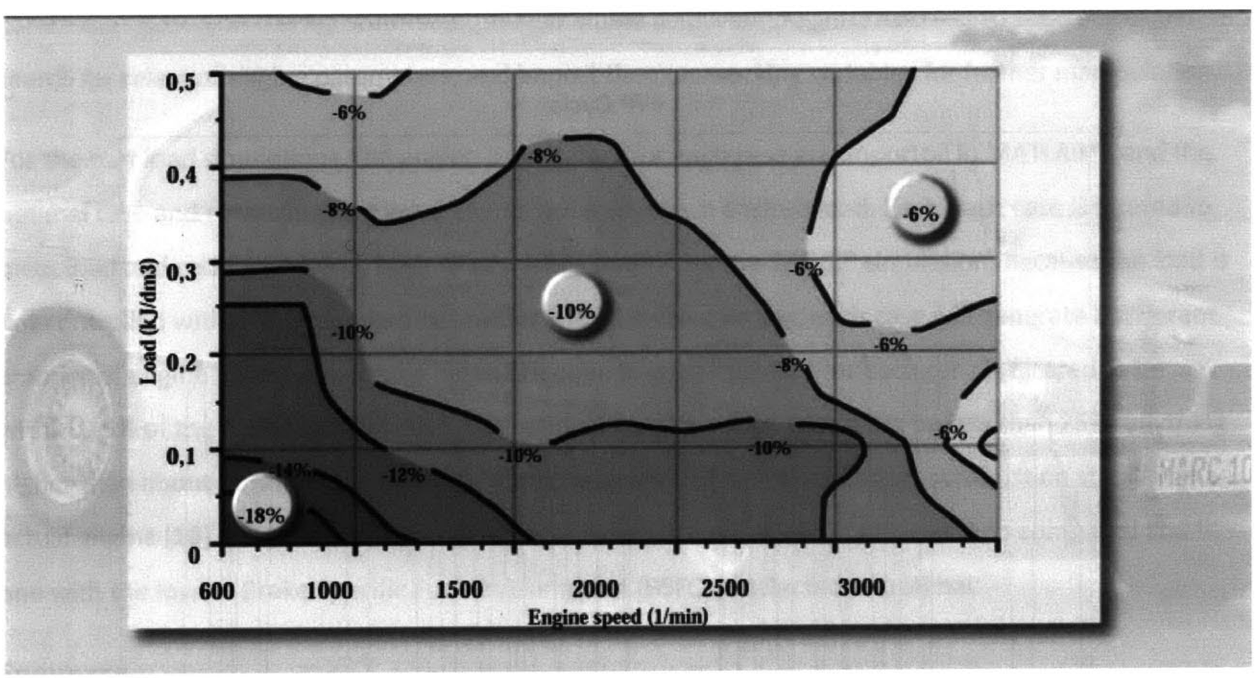

Fig. 4-2: Map-dependent consumption advantages for BMW's Valvetronic VVT [**10**]

Valve timing optimization

With the desired engine operating points selected, the next step was to determine a method for varying the valve timings to find the one that produces the optimal fuel consumption subject to the maximum residual limit of 25%. As mentioned, the computing time to run every valve configuration, even at only seven operating points, would take far too long.

The first option for finding optimal valve timings is to initially vary intake opening to find the minimum BSFC point, followed by then varying the closing time and again finding the minimum BSFC, repeating this process until variations result in no change in the minimum BSFC. The initial valve timing that is used is that of the reference engine, opening at 330° CA with a duration of 280° CA. This method is similar to that used by Assanis and Bolton [6]. One concern with this method, though, is that there may be other combinations of valve timings that are completely ignored. For example, when throttling with valve timing, the valve can either be closed very early or very late. Both options will result in the desired load, but one will have lower BSFC and it may not be the one for which the method converged. To compare these two (or possibly more) options for valve timing, a seperate method was used.

The prefered method for ranging the valve timing was to vary both opening and duration together, using slightly larger intervals at first and then refining with smaller intervals. Initially, valve opening is varied from 200° CA to 400° CA in 5° increments. For each opening point, valve duration is varied from 50° CA to 400° CA in 5° increments. This set of valve timings provides a wide range of possibilities for finding the optimal configuration. The simulation is then run for each of the generated cases except any of the valve combinations that did not meet the the valve reduction or spark timing criteria mentioned previously. The results are then analyzed to determine which combinations that lie within 2% of the desired load result in the lowest BSFC. Once the optimal timing is found, a more refined discritization of timing is used centered around that point. The intake opening is varied ±10° CA in 0.5° increments, and for each opening, the duration is also varied ±10° CA in 0.5° increments. The simulation results of the refined set are then analyzed to determine which set of valve timings that lies within 0.5% of the desired load results in the lowest BSFC. This set is then determined to be the optimal valve timing which defines the fuel consumption improvement for that particular operating point.

Both methods for determining the optimal valve timing were implemented for comparison. When special consideration was made for the first method regarding multiple optimal possibilities, the two methods were found to converge to the same valve timings. Therefore, either method can be used and the second one was chosen for robustness and simplicity.

4.5 Summary

The interface between MATLAB® and WAVE® allows for any desired valve combination to be simulated and analyzed. For a given operating point defined by engine speed and load, a set of time varied valve openings, each with a set of time varied valve durations, is used to determine the particular valve timing which optimizes fuel consumption. To determine the overall benefit of the MIT EMV, seven operating points, summarized below in Table 4-2, are chosen for optimization. The results of the simulations and optimizations are discussed in the following chapter.

Table 4-2: Operating points to be used for simulation of the MIT EMV

	Engine Speed (rpm)	Load (bar BMEP)
Chrysler	1600	2.41
Ford	1500	2.62
GM	1300	2.95
Old Standard	1500	2.5
BMW	700	0.5
	1700	2.5
	2800	3.5

Chapter 5 Results and Discussion

5.1 Introduction

This chapter discusses the results of the four corporate operating point simulations and the three BMW operating point simulations that were proposed in the previous chapter. The corporate operating point simulations were run in three different cases: Increased Lift-High Residual, Increased Lift-Limited Residual, and Standard Lift-Limited Residual; while the BMW operating points were run only for the Standard Lift-Limited Residual case. The optimized valve timings and most efficient BSFC for the MIT EMV and reference engines are compared at each operating point, and the percent improvement in fuel consumption is used as the primary performance metric. After the results are given, the physical reasoning and theoretical justification of the improvements are briefly discussed.

5.2 Results: Optimal Valve Timings and Performance Comparison

The experiment was split into two sets of simulations, the corporate operating points and the BMW operating points. The corporate operating points were run first and therefore included some additional types of simulations to determine how best to model and simulate the MIT EMV. After the best method was determined, the BMW operating points were only simulated using that one method alone.

Corporate Operating Points

Two questions arose when determining how to model and simulate the MIT EMV. First, does the lift of the MIT EMV need to be increased to make it comparable to the lift of the reference engine valve? Second, what effect does limiting the residual gas have on the optimal fuel consumption result? These two questions were addressed by running three different simulations: Increased Lift-High Residual, Increased Lift-Limited Residual, and Standard Lift-Limited Residual. The results for running each of these three simulations along with the reference engine simulation at each of the four operating points is given in Table 5-1 and are discussed in what follows.

Table 5-1: Simulation results for the four corporate operating points.
Each is run for three different scenarios plus the reference engine case. The optimized result, percent improvement in BSFC, is shown in bold.

Simulation Type	Engine Speed [rpm]	BMEP [bar]	Error relative to desired load	Intake Opening Point [deg CA]	Intake Duration [deg CA]	Throttle Angle [deg]	% Residual	BSFC [kg/kW/hr]	% Improvement in BSFC
Reference Engine	1600	2.416	0.24%	330	280	11.9	22.589	0.332	0.000
Increased Lift-High Residual	1600	2.415	0.22%	244	344	90.0	62.914	0.270	**18.648**
Increased Lift-Limited Residual	1600	2.419	0.36%	326	339	90.0	19.058	0.298	**10.211**
Standard Lift-Limited Residual	1600	2.420	0.39%	310	353	90.0	23.249	0.301	**9.456**
Reference Engine	1500	2.619	0.02%	330	280	11.8	22.193	0.324	0.000
Increased Lift-High Residual	1500	2.617	0.10%	190	395	90.0	60.993	0.267	**17.762**
Increased Lift-Limited Residual	1500	2.631	0.42%	311	347	90.0	24.862	0.285	**12.015**
Standard Lift-Limited Residual	1500	2.617	0.11%	312	349	90.0	22.574	0.290	**10.729**
Reference Engine	1300	2.942	0.26%	330	280	11.4	23.733	0.306	0.000
Increased Lift-High Residual	1300	2.953	0.09%	271	359	90.0	41.724	0.271	**11.523**
Increased Lift-Limited Residual	1300	2.956	0.19%	323	330	90.0	22.204	0.278	**9.223**
Standard Lift-Limited Residual	1300	2.961	0.38%	323	332	90.0	20.687	0.283	**7.537**
Reference Engine	1500	2.496	0.14%	330	280	11.6	22.772	0.330	0.000
Increased Lift-High Residual	1500	2.503	0.14%	250	360	90.0	57.075	0.269	**18.322**
Increased Lift-Limited Residual	1500	2.511	0.44%	315	347	90.0	21.386	0.293	**11.125**
Standard Lift-Limited Residual	1500	2.494	0.23%	314	350	90.0	21.035	0.296	**10.236**

MIT EMV Performance: Standard Lift-Limited Residual

The Standard Lift-Limited Residual simulation uses the standard 8 mm lift for the MIT EMV and also only considers results that have a residual gas less than 25%. Profiles that were run through WAVE® which resulted in a residual higher than 25% are excluded from consideration for minimum resultant BSFC. This simulation type best represents how the MIT EMV would operate if it were physically attached to an engine in its current form.

Improvements in fuel consumption are quite substantial for each operating point, ranging between 7.5% and 10.25%. The desired valve timings to achieve this benefit are each around 310° CA for intake open and 350° CA for duration, the exhaust valve timings are still fixed at 105° CA opening with a duration of 300° CA. This valve timing and the reference valve timing are shown in Fig. 5-1. Compared to the reference engine, the valve opens a bit earlier but stays open much longer. This valve timing allows the MIT EMV engine to have the throttle plate fully opened and still produce the same amount of power as the reference engine which has the throttle plate closed to around 11°-12°.

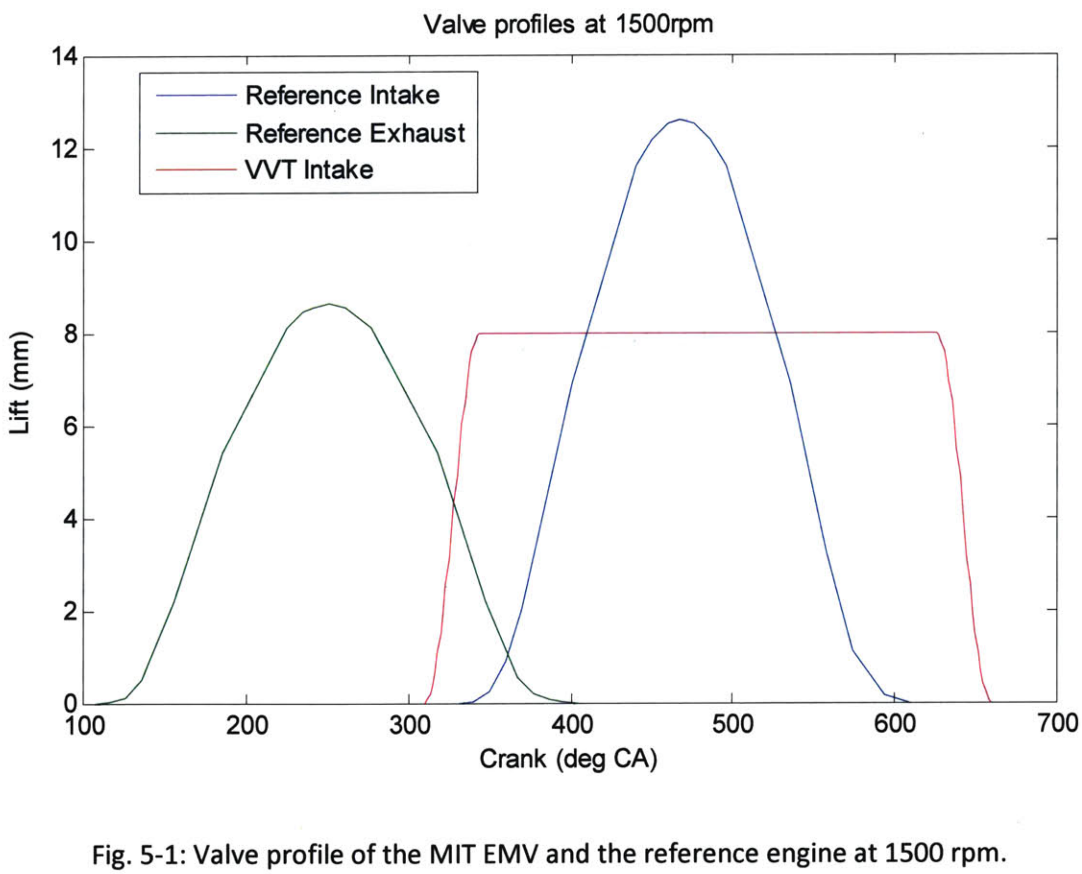

Fig. 5-1: Valve profile of the MIT EMV and the reference engine at 1500 rpm.
The MIT EMV opens at 310° CA and closes at 350° CA.

The improvement in fuel consumption comes from the reduction in pumping losses when pulling air into the cylinder. The effective displacement of the engine is changed using valve timing, which eliminates the need for a throttle plate. For the reference engine, work is done by the downward motion of the piston to pull in air through the throttle plate. With the MIT EMV, the throttle plate is wide open, so the work needed to pull the air into the cylinder is significantly reduced. The pressure drop across the throttle plate is usually necessary to limit the amount of air and fuel going into the cylinder which then controls the engine load. Instead, the duration of the MIT EMV controls the load by keeping the valve open while the piston pulls air into the cylinder and then pushes it back out into the intake manifold. The valve closes when just enough air and fuel remain in the cylinder to produce the desired load.

Effect of Increased Lift: Increased Lift-Limited Residual vs. *Standard Lift-Limited Residual*

The Increased Lift-Limited Residual simulation uses a lift multiplier of 1.571 to increase the MIT EMV model's lift to match the 12.57 mm lift of the reference engine. The residual gas is still limited to only 25%. This simulation type represents how the MIT EMV would perform under actual engine operation, but with a redesign in lift to match the reference engine.

Compared to the standard lift case, the increased lift adds from 0.7% to 1.7% to the benefit in fuel consumption over the reference engine. This improvement is due to the additional area for air to flow into and out of the cylinder. Even though increasing the MIT EMV lift to 12.57 mm would result in a slightly lower fuel consumption in simulation, such a high valve lift is unconventional in practice. Typical maximum valve lift to diameter ratios are 0.25 [**18**]. The valve diameter is 28 mm which would set the maximum lift at 7 mm by convention. Therefore, in spite of the slight improvements seen in the simulation, the previously designed lift of 8 mm is more appropriate for the MIT EMV in practice.

Effect of Residual Gas: Increased Lift-High Residual vs. Increased Lift-Limited Residual

The Increased Lift-High Residual simulation also uses a lift multiplier of 1.571 to increase the MIT EMV model's lift, but it has no limitation on residual gas; any profile that is run through WAVE® is considered for minimizing BSFC. This simulation type represents the best case scenario of the MIT EMV when compared to the reference engine.

It is shown that the improvement in fuel consumption from the high residual cases is approximately 1.25 to 2 times higher than those from the limited residual cases. This improvement, however, is likely to be seen only in simulation and would not exist in actual implementation. Residual gas refers to exhaust gas left in the cylinder which displaces the fresh charge coming in. Too much residual causes unstable combustion and high cycle to cycle variations. WAVE® does not consider this combustion instability and will still simulate the engine cycle. The simulated engine produces a high amount of power, but with a limited fuel input, resulting in a low BSFC. While these simulation results give some insight on how residual affects the simulation, they are not useful for determining the performance of the MIT EMV due to the unrealistic combustion performance.

BMW Operating points

Simulations were also run and BSFC was optimized at three operating points which correspond to those that were published by BMW with respect to their Valvetronic variable valve actuator. The simulations for these BMW operating points are similar to those of the corporate operating points, except that the increased lift and high residual cases are not included due to their impractical nature as discussed. The one exception being a higher residual case which was run for the 700 rpm and 0.5 bar operation point. At this point, when optimizing the BSFC with the standard 25% residual limit, it was seen that fuel consumption achieved with the MIT EMV could not improve upon that of the reference engine at wide open throttle, and the improvement is in fact negative. This specific operating point corresponds to engine idle, where typically the exhaust residual is quite large due to the low engine speed and high vacuum in the intake manifold [22]. Combustion at such a high residual is once again unlikely, but such is the output of the reference engine. Therefore for comparison, the residual limit for the MIT EMV simulation was modified so that the percent residual gas could be as high as that of the reference engine, up to 61%.

The results from the simulations of these three operating points are shown in Table 5-2. For reference, the fuel consumption improvements published by BMW are summarized in Table 5-3. Comparing the results of the MIT EMV to those of the BMW Valvetronic valve drive, the percent improvements at each operating point are remarkably similar. Both show a trend that as load and speed increase, the improvement in fuel consumption is decreased. This is to be expected because the throttle related pumping losses become smaller as the load is increased. Also, the ability to vary valve duration significantly is reduced as engine speed increases.

Table 5-2: Simulation results for the three operating points from BMW's published results. The optimized results, percent improvement in BSFC, are shown in bold.

Simulation Type	Engine Speed [rpm]	BMEP [bar]	Error relative to desired load	Intake Opening Point [deg CA]	Intake Duration [deg CA]	Throttle Angle [deg]	% Residual	BSFC [kg/kW/hr]	% Improvement in BSFC
Reference Engine	700	0.495	0.98%	330	280	6.9	61.641	0.690	0.000
Standard Lift-Limited Residual	700	0.505	0.95%	364	325	90.0	22.168	0.724	**-4.927**
Standard Lift-High Residual	700	0.502	0.38%	263	159	90.0	59.686	0.567	**17.841**
Reference Engine	1700	2.503	0.12%	330	280	12.3	21.918	0.327	0.000
Standard Lift-Limited Residual	1700	2.511	0.43%	317	346	90.0	20.793	0.297	**9.204**
Reference Engine	2800	3.516	0.45%	330	280	16.5	15.887	0.290	0.000
Standard Lift-Limited Residual	2800	3.509	0.25%	271	381	90.0	24.442	0.271	**6.587**

Table 5-3: Summary of percent improvement reported by BMW using Valvetronic [**10**].

Engine Speed [rpm]	BMEP [bar]	% Improvement in BSFC
700	0.5	18%
1700	2.5	10%
2800	3.5	6%

5.3 Summary

The results show that the MIT EMV can achieve improvements in fuel consumption at each specified operating point. There is an improvement of about 10% at each of the corporate operating points which shows to the automotive companies that the MIT EMV can be beneficial in their engines. Likewise, improvements of approximately 18%, 9%, and 6.6% were seen in the MIT EMV at the BMW operating points. These improvements are well aligned with BMW Valvetronic's improvements of 18%, 10%, and 6%, showing that the MIT EMV, using only valve timing to control load, is capable of achieving the same performance benefits as similar leading actuators on the market which also have the capability of varying valve lift to control engine load.

Additional simulations where the maximum valve lift of the MIT EMV was increased showed some improvement, but were determined to be impractical due to the high valve lift to diameter ratio produced. Similarly, the impact of high residual was explored and significant improvement was shown, but this improvement was also deemed impractical due to the negative impacts of high residual on combustion stability and cycle to cycle variations in physical engines.

Chapter 6 Conclusion

6.1 Introduction

This chapter concludes the thesis with the evaluation of the thesis goals followed by suggestions for future work.

6.2 Evaluation of Thesis Objectives

As mentioned in Chapter 1, the primary goal of this thesis is to determine quantitatively the benefits in fuel efficiency that the MIT EMV provides and the following objectives were purposed to meet this thesis goal:

1. Determine a reference engine with which to evaluate the MIT EMV
2. Find a method with which to simulate engine performance
3. Model the reference engine and the MIT EMV within the engine simulation software
4. Determine a simulation strategy to best evaluate potential performance gains
5. Perform simulations and then review and discuss the results

Chapter 3 provides an explanation for the choice in reference engine along with the obtaining of the model of the engine from WAVE®. While a reference engine from industry would possibly be more useful, the engine from the WAVE® model was determined to be sufficient and reduces the modeling challenges. In the same chapter, the modeling of the MIT EMV based on the parameters of the physical drive is also discussed.

Chapter 4 discusses the choice of WAVE® as the ideal method for simulating engine performance. Following that, the choice of simulation strategy and the use of corporate and BMW operating points to best evaluate performance gains are discussed.

Finally, Chapter 5 gives the results of the performed simulations and offers a discussion on the meaning and physical reasoning for those results. The results indicate that the MIT EMV is indeed capable of reducing fuel consumption by approximately 10% at specific operating points that correspond to the performance of an engine on the FTP cycle. The actuator is also capable of achieving benefits of 18%, 9%, and 6.6% which correspond to the same improvements seen by the BMW Valvetronic actuator at another set of specific operating points. Thus, the thesis goal has been successfully completed.

6.3 Recommendations for the Future

Ideally, it would be great to have the MIT EMV implemented on an actual physical engine and the work of this thesis has provided some reference on how to operate the valve in such a setup. Unfortunately, even though much work has been done on the physical design of the MIT EMV, there is still a lot that should be done to prepare it for physical experimentation. Dr. Qiu's thesis covers in detail these issues of improving control, nonlinear implementation, actuation, position sensing, lash adjustment, and cooling [2]; so they will not be discussed here. The discussion will focus instead on what more work could be done on simulating the MIT EMV.

The simulations for this project focused on showing that the MIT EMV can indeed improve fuel efficiency and that it can compete with other actuators. To further expand on this work, it is recommended that a more complete engine map be made. This would require determining optimal fuel efficiencies at many more speed and load points. Such a task requires much more time, but now that a methodology for easy simulation has been implemented, the task could be automated. Accomplishing this task would not only give a better idea on the overall performance of the MIT EMV, but it would give specific valve timings which could be used when the MIT EMV is implemented on a physical engine.

As mentioned in Chapter 2, variable valve timing can perform more functions than just valve throttling. The performance of the MIT EMV in performing engine braking, exhaust residual control, cylinder deactivation, multi-valve control, and better breathing could all be explored using WAVE®. If the MIT EMV were to be placed on a production engine, each of these aspects should be utilized, and determining how best to do that is a great task for engine simulation software.

Appendix I MATLAB® Program for Generating WAVE® simulations

The following three files are used to generate the data for a reference engine simulation, the data for an MIT EMV simulation, or the data for specified MIT EMV valve lift profile.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MIT EMV Evaluation                                              %
% Justin Miller                                                   %
% May 2011                                                        %
% Name of file: ref_table.m                                       %
% Purpose: Script which uses the data from the reference engine model to %
% develop that same data for several throttle settings at any engine     %
% speed through interpolation                                     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Speed dependant data taken from the WAVE® reference engine model
A_F = [ 14.7    14.7    14.7    14.7    14.7    14.7    14.7    14.7];
BDUR = [33  32  31  32  31  29.5    29  28];
CA50 = [7.5 7.5 8   8.5 8.5 9   9   9   ];
CAT_TEMP = [1135 1120 1075 1025 950 900 830 790];
EV_TEMP = [388  383 380 377 375 370 366 365];
EXH_OPEN =  [105    105 105 105 105 105 105 105 105 105 105 105 105 105];
HEAD_TEMP = [ 648   639 635 630 620 595 580 550 ];
INT_OPEN = [330 330 330 330 330 330 330 330 330 330 330 330 330 330];
IV_TEMP = [ 325 322 320 320 320 318 316 312 ];
LINER_TEMP =  [628  622 615 600 595 580 570 540];
PISTON_TEMP = [609  603 595 585 580 550 530 500 ];
RUNNER_LENGTH   = [188  188 188 188 188 188 188 188];
SPEED = [ 8000  7000    6000    5000    4000    3000    2000    1000];

%Assign a vector for each throttle value to be analized
throttle = [90 75 50 25:-0.1:15.1 15:-0.01:10  9:-0.1:0];

%Express the desired engine speed
des_speed = 2800;

%Determine whther to use interpolation or extrapolation (for speeds under
%1000 rpm). Then find the index of the two values closest to des_speed
if des_speed <1000
    a = 8;
    b = 7;
else
    a = find(SPEED==floor(des_speed/1000)*1000);
    b = find(SPEED==ceil(des_speed/1000)*1000);
end

%Linear interpolation multiplier
mult = (des_speed - SPEED(b))/(SPEED(a)-SPEED(b));

%Linearly interpolate each parameter and write it to a matrix, do this for
%each throttle angle
```

```
    for j = 1:length(throttle)
        idx = j;
        M(1,idx) = (A_F(a)*mult+A_F(b)*(1-mult));
        M(2, idx) = (BDUR(a)*mult+BDUR(b)*(1-mult));
        M(3, idx) = (CA50(a)*mult+CA50(b)*(1-mult));
        M(4, idx) = (CAT_TEMP(a)*mult+CAT_TEMP(b)*(1-mult));
        M(5, idx) = (EV_TEMP(a)*mult+EV_TEMP(b)*(1-mult));
        M(6, idx) = (EXH_OPEN(a)*mult+EXH_OPEN(b)*(1-mult));
        M(7, idx) = (HEAD_TEMP(a)*mult+HEAD_TEMP(b)*(1-mult));
        M(8, idx) = (INT_OPEN(a)*mult+INT_OPEN(b)*(1-mult));
        M(9, idx) = (IV_TEMP(a)*mult+IV_TEMP(b)*(1-mult));
        M(10, idx) = (LINER_TEMP(a)*mult+LINER_TEMP(b)*(1-mult));
        M(11, idx) = (PISTON_TEMP(a)*mult+PISTON_TEMP(b)*(1-mult));
        M(12, idx) = (RUNNER_LENGTH(a)*mult+RUNNER_LENGTH(b)*(1-mult));
        M(13, idx) = (SPEED(a)*mult+SPEED(b)*(1-mult));
        M(14, idx) = throttle(j);
    end

%Write the matrix as an excel filie to be transfered directly to WAVE®.
delete('ref_table.xlsx');
[SUCCESS,MESSAGE] = xlswrite('ref_table.xlsx',M)

%Display then number of runs for the simulation to be used in posti_split.m
idx
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MIT EMV Evaluation                                                      %
% Justin Miller                                                           %
% May 2011                                                                %
% Name of file: tablei_split.m                                            %
% Purpose: Script which uses the data from the reference engine model to  %
% develop that same data for several intake valve timings at any engine   %
% speed through interpolation                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Specify the range of intake opening points and the number of intake
%durations at each of those intake points
int_open = [200:10:400] ;
int_dur = [50:10:400] ;

%Set throttle angle, all cases are 90 degrees at WOT
throttle = 90;

%Express the desired engine speed
des_speed = 2800;

%Speed dependant data taken from the WAVE® reference engine model
A_F = [ 14.7    14.7    14.7    14.7    14.7    14.7    14.7    14.7];
BDUR = [33  32  31  32  31  29.5    29  28];
CA50 = [7.5 7.5 8   8.5 8.5 9   9   9   ];
CAT_TEMP = [1135 1120 1075 1025 950 900 830 790];
EV_TEMP = [388  383 380 377 375 370 366 365];
HEAD_TEMP = [ 648   639 635 630 620 595 580 550];
IV_TEMP = [ 325 322 320 320 320 318 316 312];
LINER_TEMP =  [628  622 615 600 595 580 570 540];
PISTON_TEMP = [609  603 595 585 580 550 530 500];
RUNNER_LENGTH   = [188  188 188 188 188 188 188 188];
SPEED = [ 8000  7000    6000    5000    4000    3000    2000    1000];

%Determine whther to use interpolation or extrapolation (for speeds under
%1000 rpm). Then find the index of the two values closest to des_speed
if des_speed <1000
    a = 8;
    b = 7;
else
    a = find(SPEED==floor(des_speed/1000)*1000);
    b = find(SPEED==ceil(des_speed/1000)*1000);
end

%Linear interpolation multiplier
mult = (des_speed-SPEED(b))/(SPEED(a)-SPEED(b));

%Loop through each throttle setting, intake opening point, and intake
%duration. Build the valve lift profile for each. Then, linearly interpolate
%the remaining engine parameters and write everything to a matrix.
l = length(int_open);
m = length(int_dur);
n = length(SPEED);
idx = 1;
for k=1:length(throttle)
```

```
    for j = 1:m
        for i = 1:l
            [CA,z,lift] = valveprofile(int_open(i),int_dur(j),des_speed);

            %Ensure the valve will not be open when spark is initiated.
            if CA(end)>=(CA50(a)*mult+CA50(b)*(1-mult))-
(BDUR(a)*mult+BDUR(b)*(1-mult))+715
                    display(['Warning intake valve is open when spark occurs for
IVO = ' num2str(int_open(i)) ', IV_Dur = ' num2str(int_dur(j)) ' and RPM = '
num2str((SPEED(a)+SPEED(b))*mult)])

            %Ensure the maximum valve lift is not reduced by more than 2%.
            elseif lift<(0.98*8)
                    display(['Maximum Intake Lift = ' num2str(lift) 'mm'])

           %Only if both conditions are met, proceed with creating the matrix
            else
                M(1:44,idx) = CA;
                M(45,idx) = (A_F(a)*mult+A_F(b)*(1-mult));
                M(46, idx) = (BDUR(a)*mult+BDUR(b)*(1-mult));
                M(47, idx) = (CA50(a)*mult+CA50(b)*(1-mult));
                M(48, idx) = (CAT_TEMP(a)*mult+CAT_TEMP(b)*(1-mult));
                M(49, idx) = (EV_TEMP(a)*mult+EV_TEMP(b)*(1-mult));
                M(50, idx) = (HEAD_TEMP(a)*mult+HEAD_TEMP(b)*(1-mult));
                M(51, idx) = (IV_TEMP(a)*mult+IV_TEMP(b)*(1-mult));
                M(52:95,idx) = z;
                M(96, idx) = (LINER_TEMP(a)*mult+LINER_TEMP(b)*(1-mult));
                M(97, idx) = (PISTON_TEMP(a)*mult+PISTON_TEMP(b)*(1-mult));
                M(98, idx) = (RUNNER_LENGTH(a)*mult+RUNNER_LENGTH(b)*(1-
mult));
                M(99, idx) = (SPEED(a)*mult+SPEED(b)*(1-mult));
                M(100, idx) =   throttle(k);
                idx = idx+1;
            end
        end
    end
end

%Write the matrix into four files to be run in four simultaneous simulations
split = floor(idx/4)

delete('tablei_1.xlsx');
[SUCCESS,MESSAGE] = xlswrite('tablei_1.xlsx',M(:,1:split));
MESSAGE
delete('tablei_2.xlsx');
[SUCCESS,MESSAGE] = xlswrite('tablei_2.xlsx',M(:,split+1:2*split));
MESSAGE
delete('tablei_3.xlsx');
[SUCCESS,MESSAGE] = xlswrite('tablei_3.xlsx',M(:,2*split+1:3*split));
MESSAGE
delete('tablei_4.xlsx');
[SUCCESS,MESSAGE] = xlswrite('tablei_4.xlsx',M(:,3*split+1:end));
MESSAGE

%Display the number of total runs for use in posti_split.m
idx-1
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MIT EMV Evaluation                                                         %
% Justin Miller                                                              %
% May 2011                                                                   %
% Name of file: valveprofile.m                                               %
% Purpose: Function which creates a valve lift profile defined by crank     %
% angle vs. lift points based on a specified intake opening and duration    %
% and engine speed.                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [CA,z,lift] = valveprofile(ivo,duration,N,dummie)
%"ivo", "duration", and "N" are the specified opening, duration, and speed,
respectively
%"dummie" is an option to specify if the profile should be plotted
%"CA" and "z" are vectors of corresponding crank angle and lift points,
respectively
%"lift" is the maximum lift of the profile

th1 = 1/6*asind((0.05*8-4)/4*sin(0.999*pi()/2));             % Angle of 5%
valve lift
th2 = 1/6*asind((0.95*8-4)/4*sin(0.999*pi()/2));             % Angle of 95%
valve lift



total_time = duration/(6*N);                                 %Amount of time
the valve will be open
if total_time > 8.3e-3+2*(th1+15)*2.7e-3/(th2-th1);          %If it is greater
than the 8.3ms case (plus added time for 0 to 5%)
    th_stop = 15;
elseif total_time > 2*(15+15)*2.7e-3/(th2-th1)               %If it is less
than the 8.3ms case but slower than back to back valve events
    m = (1-0.98)/(8.3e-3-6.8e-3);                            % m,b, and x are
used to calculate %lift linear interpolation between 8.3ms and 6.3ms
    b = 1-m*8.3e-3;
    x = total_time-2*(th1+15)*2.7e-3/(th2-th1);
    th_stop = 1/6*asind(((m*x+b)*8-4)/4*sin(0.999*pi()/2)); %Theta that
generates the desired %lift
else
    th_stop = total_time/2*(th2-th1)/2.7e-3 - 15;            %Otherwise just
do back to back transitions with no delay
end

time_stop = (th_stop+15)*2.7e-3/(th2-th1);                   %Calculate the
time it takes for the valve to open from zero

theta = linspace(-15,th_stop,22);                            %Build angle and
time vecotrs

time = linspace(0,time_stop,length(theta));

for i=1:length(theta)                                        %Determine the
trajectory of the valve opening
    if abs(theta(i))<15
    z(i) = 4*sind(6*theta(i))/sin(0.999*pi/2)+4;
    else
```

```
    z(i) = 4*sign(theta(i))+4;
    end
end

lift = z(end);                                              %Record max lift
for output

z = [z fliplr(z)]';                                         %Build lift
vector
CA = [time*6*N+ivo, time*6*N+ivo+(duration-time_stop*6*N)]';%Build CA vector
with enough gap to ensure proper duration

if CA(23) == CA(22);                                        %WAVE® cannot
accept two valve timing settings at the same Crank Angle, so
    CA(23) = CA(23)+0.01;                                   %a 0.01 degree
offset is added
end

if nargin==4                                                %Plot the
function if a dummie variable is specified
figure
plot(CA,z)
title(['Lift vs. Crank Angle at ',num2str(N),'rpm'])
xlabel('Crank angle (deg)')
ylabel('Lift (mm)')
axis([CA(1)-50 CA(end)+50 z(1) z(length(z)/2)+1])
end
```

Appendix II MATLAB® Program for reading WAVE® simulations

The following three files are used to read the data from a reference engine and a MIT EMV simulation, determine the valve timing which optimizes BSFC, and export the data to a results table.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MIT EMV Evaluation                                                    %
% Justin Miller                                                         %
% May 2011                                                              %
% Name of file: posti_split.m                                           %
% Purpose: Script which reads the reference engine and MIT EMV simulations%
% which were run at a given speed, determines the values within each that %
% meet the specified engine load, and returns valve timing information   %
% about the MIT EMV that provides the most fuel efficiency.             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ccc

num_of_runs = 1681; %Number of runs in the MIT EMV simulation

OP = 3.5;           %Operating point, bar BMEP

rpms = 1;           %Number of different engine speeds in the simulation

throttles = 695;    %Number of runs in the reference engine simulation

tol = 0.5;          %The percent that the actual BMEP of the MIT EMV
                    %must be relative to the desired BMEP

res = 1;            % To ignore residual < 30 constraint, set res to 0

long_dur = 1;       % To exclude durations shorter than 200 deg CA set dur to
0

%--Reference-Engine--------------------------------------------------
%Specifiy strings to search within the simulation ".sum" file, "e" at the
%end refers to the fact that the search ends at that string
rpms_ref = rpms;
str1_ref = ' RPM= ';
str1e_ref = 'PAMBE= ';
str2_ref = 'BHP= ';
str2e_ref = 'BPOWKW= ';
str3_ref = 'INT_OPEN= ';
str3e_ref = 'IV_TEMP= ';
str4_ref = 'EXH_OPEN= ';
str4e_ref = 'HEAD_TEMP= ';
str5_ref = 'BSFCSI= ';
str5e_ref = 'BSFCBR=';
str6_ref = 'THROTTLE_ANGLE= ';
str6e_ref = '#';
str7_ref = 'PMEPSI_PX= ';
```

```
str7e_ref = 'FMEPSI= ';

str8_ref = 'BSHC= ';
str8e_ref = 'PPMCO= ';
str9_ref = 'BSNO2= ';
str9e_ref = 'PPMHC= ';
str10_ref = 'BSCO= ';
str10e_ref = 'HTCIVP1= ';
str11_ref = 'RES%= ';
str11e_ref = 'VOLEFD= ';
str12_ref = 'BMEPSI= ';
str12e_ref = 'BMEPBR= ';
str13_ref = 'IMEPSI= ';
str13e_ref = 'IMEPBR= ';
str14_ref = 'FMEPSI= ';
str14e_ref = 'FMEPBR= ';

%Open the ".sum" file and scan it into MATLAB®
file_ref = 'tut_si4_16v_ref.sum';
fid_ref=fopen(file_ref,'r');
d_ref=fscanf(fid_ref,'%c');
st_ref = fclose(fid_ref);

%Find the locations of each string specified within the scanned file
k1_ref = findstr(str1_ref, d_ref);
k2_ref = findstr(str2_ref, d_ref);
k3_ref = findstr(str3_ref, d_ref);
k4_ref = findstr(str4_ref, d_ref);
k5_ref = findstr(str5_ref, d_ref);
k6_ref = findstr(str6_ref, d_ref);
k7_ref = findstr(str7_ref, d_ref);
k8_ref = findstr(str8_ref, d_ref);
k9_ref = findstr(str9_ref, d_ref);
k10_ref = findstr(str10_ref, d_ref);
k11_ref = findstr(str11_ref, d_ref);
k12_ref = findstr(str12_ref, d_ref);
k1e_ref = findstr(str1e_ref, d_ref);
k2e_ref = findstr(str2e_ref, d_ref);
k3e_ref = findstr(str3e_ref, d_ref);
k4e_ref = findstr(str4e_ref, d_ref);
k5e_ref = findstr(str5e_ref, d_ref);
k6e_ref = findstr(str6e_ref, d_ref);
k7e_ref = findstr(str7e_ref, d_ref);
k8e_ref = findstr(str8e_ref, d_ref);
k9e_ref = findstr(str9e_ref, d_ref);
k10e_ref = findstr(str10e_ref, d_ref);
k11e_ref = findstr(str11e_ref, d_ref);
k12e_ref = findstr(str12e_ref, d_ref);
k13_ref = findstr(str13_ref, d_ref);
k13e_ref = findstr(str13e_ref, d_ref);
k14_ref = findstr(str14_ref, d_ref);
k14e_ref = findstr(str14e_ref, d_ref);

%Check to make sure the number of found strings is the same as the number
%of cases
if length(k1_ref) ~= throttles
```

```
    display(['actual throttles = ' num2str(length(k1_ref)) ', specified = '
num2str(throttles)])
    break
end

%For each instance of each string, pull the value from the scanned file and
%save it in a vector associated with what the value refers to
for i=1:rpms
    for j=1:throttles %Number of throttle settings
        idx_ref = (i-1)*throttles + j;

        if isvector(k1_ref)>0
            PMEP_ref(j) =
str2double(d_ref(k7_ref(idx_ref)+11:k7e_ref(idx_ref)-1));
            Throttle_ref(j) =
str2double(d_ref(k6_ref(idx_ref)+16:k6e_ref(idx_ref)-1));
            RPM_ref(j) = str2double(d_ref(k1_ref(idx_ref)+9:k1e_ref(idx_ref)-
1));
            BHP_ref(j) = str2double(d_ref(k2_ref(idx_ref)+9:k2e_ref(idx_ref)-
1));
            INT_OPEN_ref(j) =
str2double(d_ref(k3_ref(idx_ref)+9:k3e_ref(idx_ref)-1));
            EXH_OPEN_ref(j) =
str2double(d_ref(k4_ref(idx_ref)+9:k4e_ref(idx_ref)-1));
            BSFC_ref(j) =
str2double(d_ref(k5_ref(idx_ref)+9:k5e_ref(idx_ref)-1));
            BSHC_ref(j) =
str2double(d_ref(k8_ref(idx_ref)+6:k8e_ref(idx_ref)-1));
            BSNO2_ref(j) =
str2double(d_ref(k9_ref(idx_ref)+7:k9e_ref(idx_ref)-1));
            BSCO_ref(j) =
str2double(d_ref(k10_ref(idx_ref)+6:k10e_ref(idx_ref)-1));
            RES_ref(j) =
str2double(d_ref(k11_ref(idx_ref)+5:k11e_ref(idx_ref)-1));
            BMEP_ref(j) =
str2double(d_ref(k12_ref(idx_ref)+8:k12e_ref(idx_ref)-1));
            IMEP_ref(j) =
str2double(d_ref(k13_ref(idx_ref)+8:k13e_ref(idx_ref)-1));
            FMEP_ref(j) =
str2double(d_ref(k14_ref(idx_ref)+8:k14e_ref(idx_ref)-1));
        else
        display('error')
        end
    end
end

%Determine the value in BMEP which is closest to the desired BMEP operating
%point, then determine all of the other associated engine properties at
%that point
[BMEP_ref_OP,loc_OP,err_OP] = closevalue(OP,BMEP_ref);
RPM_ref_OP= RPM_ref(loc_OP');
INT_OPEN_ref_OP= INT_OPEN_ref(loc_OP');
EXH_OPEN_ref_OP= EXH_OPEN_ref(loc_OP');
BSFC_ref_OP= BSFC_ref(loc_OP');
Throttle_ref_OP= Throttle_ref(loc_OP');
PMEP_ref_OP= PMEP_ref(loc_OP');
```

```
BSHC_ref_OP= BSHC_ref(loc_OP');
BSNO2_ref_OP= BSNO2_ref(loc_OP');
BSCO_ref_OP= BSCO_ref(loc_OP');
RES_ref_OP= RES_ref(loc_OP');
BHP_ref_OP= BHP_ref(loc_OP');
IMEP_ref_OP= IMEP_ref(loc_OP');
FMEP_ref_OP= FMEP_ref(loc_OP');


%--VVT-------------------------------------------------------------------
%Specifiy strings to search within the MIT EMV simulation ".sum" file,
%"e" at the end refers to the fact that the search ends at that string
str1 = ' RPM= ';
str1e = 'PAMBE= ';
str2 = 'BHP= ';
str2e = 'BPOWKW= ';
str3 = 'A001= ';
str3e = 'A002= ';
str4 = 'A044= ';
str4e = 'A_F= ';
str7 = 'BSFCSI= ';
str7e = 'BSFCBR= ';
str8 = 'PMEPSI_PX= ';
str8e = 'FMEPSI= ';

str9 = 'BSHC= ';
str9e = 'PPMCO= ';
str10 = 'BSNO2= ';
str10e = 'PPMHC= ';
str11 = 'BSCO= ';
str11e = 'HTCIVP1= ';
str12 = 'RES%= ';
str12e = 'VOLEFD= ';
str13 = 'BMEPSI= ';
str13e = 'BMEPBR= ';
str14 = 'THROTTLE_ANGLE= ';
str14e = '#';
str15 = 'IMEPSI= ';
str15e = 'IMEPBR= ';
str16 = 'FMEPSI= ';
str16e = 'FMEPBR= ';

%To speed up calculations, a single WAVE® simulation is split into 4
%simulations which are run simultaneously. Each  ".sum" file is opened and
% then scanned it into MATLAB®
file1 = 'tut_si4_16v_VVTi - 1.sum';
fid1=fopen(file1,'r');
file2 = 'tut_si4_16v_VVTi - 2.sum';
fid2=fopen(file2,'r');
file3 = 'tut_si4_16v_VVTi - 3.sum';
fid3=fopen(file3,'r');
file4 = 'tut_si4_16v_VVTi - 4.sum';
fid4=fopen(file4,'r');
d=[fscanf(fid1,'%c') fscanf(fid2,'%c') fscanf(fid3,'%c') fscanf(fid4,'%c')];
st = fclose('all');
```

```
%Find the locations of each string specified within the scanned file
q1 = findstr(str1, d);
q2 = findstr(str2, d);
q3 = findstr(str3, d);
q4 = findstr(str4, d);
q7 = findstr(str7, d);
q8 = findstr(str8, d);
q9 = findstr(str9, d);
q10 = findstr(str10, d);
q11 = findstr(str11, d);
q12 = findstr(str12, d);
q13 = findstr(str13, d);
q14 = findstr(str14, d);
q1e = findstr(str1e, d);
q2e = findstr(str2e, d);
q3e = findstr(str3e, d);
q4e = findstr(str4e, d);
q7e = findstr(str7e, d);
q8e = findstr(str8e, d);
q9e = findstr(str9e, d);
q10e = findstr(str10e, d);
q11e = findstr(str11e, d);
q12e = findstr(str12e, d);
q13e = findstr(str13e, d);
q14e = findstr(str14e, d);
q15 = findstr(str15, d);
q15e = findstr(str15e, d);
q16 = findstr(str16, d);
q16e = findstr(str16e, d);
idx = 0;

%Check to make sure the number of found strings is the same as the number
%of cases
if length(q1) ~= num_of_runs
    display(['actual runs = ' num2str(length(q1)) ', specified = '
num2str(num_of_runs)])
    break
end

%For each instance of each string, pull the value from the scanned file and
%save it in a vector associated with what the value refers to
for idx = 1:num_of_runs
     if isvector(q1)>0
        RPM(idx) = str2double(d(q1(idx)+9:q1e(idx)-1));
        BHP(idx) = str2double(d(q2(idx)+9:q2e(idx)-1));
        INT_OPEN(idx) = str2double(d(q3(idx)+9:q3e(idx)-1));
        INT_DURATION(idx) = str2double(d(q4(idx)+9:q4e(idx)-1))-
INT_OPEN(idx);
        BSFC(idx) = str2double(d(q7(idx)+9:q7e(idx)-1));
        PMEP(idx) = str2double(d(q8(idx)+11:q8e(idx)-1));
        BSHC(idx) = str2double(d(q9(idx)+6:q9e(idx)-1));
        BSNO2(idx) = str2double(d(q10(idx)+7:q10e(idx)-1));
        BSCO(idx) = str2double(d(q11(idx)+6:q11e(idx)-1));
        RES(idx) = str2double(d(q12(idx)+5:q12e(idx)-1));
        BMEP(idx) = str2double(d(q13(idx)+8:q13e(idx)-1));
        Throttle(idx) = str2double(d(q14(idx)+16:q14e(idx)-1));
```

```
        IMEP(idx) = str2double(d(q15(idx)+8:q15e(idx)-1));
        FMEP(idx) = str2double(d(q16(idx)+8:q16e(idx)-1));
    else
    display('error')
    end
end

%Go through every value of BMEP to determine which fall within the
%tolerance range specified by "tol", possibly within "res" and "long_dur"
%restrictions (see partialload.m for details). Of those that do fall in
%that range, determine which is has the lowest BSFC and assign the location
%of it as "loc7". Then determine all engine parameters at that location.
for i = 1:rpms
    [loc7,idxes7] =
partialload(tol,OP,BMEP,BSFC,RES,res,INT_DURATION,long_dur);
    if isempty(loc7)==1;
        display('BHP contains no value at OP%');
    else
        BHP_OP(i) = BHP(loc7);
        BSFC_OP(i) = BSFC(loc7);
        INT_OPEN_OP(i) = INT_OPEN(loc7);
        INT_DURATION_OP(i) = INT_DURATION(loc7);
        RPM_OP(i) = RPM(loc7);
        PMEP_OP(i) = PMEP(loc7);
        BSHC_OP(i) = BSHC(loc7);
        BSNO2_OP(i) = BSNO2(loc7);
        BSCO_OP(i) = BSCO(loc7);
        RES_OP(i) = RES(loc7);
        BMEP_OP(i) = BMEP(loc7);
        Throttle_OP(i) = Throttle(loc7);
        IMEP_OP(i) = IMEP(loc7);
        FMEP_OP(i) = FMEP(loc7);
    end
end


%Write desired reference and MIT EMV engine parameters at the lowest BSFC
%point to a csv file as the final result
M = [RPM_ref_OP BMEP_ref_OP abs(BMEP_ref_OP-OP)/OP*100 BSFC_ref_OP
INT_OPEN_ref_OP 280 BSHC_ref_OP BSNO2_ref_OP BSCO_ref_OP Throttle_ref_OP
RES_ref_OP 0;
    RPM_OP BMEP_OP abs(BMEP_OP-OP)/OP*100 BSFC_OP INT_OPEN_OP INT_DURATION_OP
BSHC_OP BSNO2_OP BSCO_OP  Throttle_OP RES_OP (BSFC_OP(1)-
BSFC_ref_OP(1))/BSFC_ref_OP(1)*100]';

dlmwrite('resultsi2.csv',M','-append')

%Display the improvement in BSFC over the reference engine
if numel(BSFC_OP)==1
display(['Percent improvement for OP: ' num2str((BSFC_OP(1)-
BSFC_ref_OP(1))/BSFC_ref_OP(1)*100) '%'])
end

%Plot the valve timing at the final result to ensure it looks reasonable
plotVVT(INT_OPEN_OP,INT_DURATION_OP,RPM_OP,INT_OPEN_ref_OP,EXH_OPEN_ref_OP)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MIT EMV Evaluation                                                    %
% Justin Miller                                                         %
% May 2011                                                              %
% Name of file: closevalue.m                                            %
% Purpose: %To find the value and location of a number in a given       %
% vector that is closest to a desired value. Used for determining the   %
% throttle angle which provides the desired BMEP in the reference engine%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [val,loc,err] = closevalue(num,vec)
%"num" is a desired value to find within the vector "vec"
%"val" is the value in that vector closest to "num"
%"loc" is the index of "val" within "vec"
%"err" is the percent error of "val" relative to "num"

d0 = inf;                           %initialize d0
for i=1:length(vec)
    diff = num - vec(i);            %check the error between num and each
elemenet
    if abs(diff)<d0
        d0 = abs(diff);             %keep the value and record the location of
        val = vec(i);               %the value which has the least error
        loc = i;
        err = abs(diff)/num*100;
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MIT EMV Evaluation                                                    %
% Justin Miller                                                         %
% May 2011                                                              %
% Name of file: partialload.m                                           %
% Purpose: Determines the location of the value in BMEP which conforms  %
% to the load range while giving the lowest BSFC. Range is percent      %
% above and below the load (i.e. 2% would be 98%-102% of BMEP_ref)      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [Location,idxes] =
partialload(range,BMEP_ref,BMEP,BSFC,RES,res,INT_DUR,long_dur)

%"range" is percent above and below the load (i.e. 2% would be 98%-102% of
%BMEP_ref), "BMEP_ref" is the desired BMEP, "BMEP" is the vector of all
engine BMEP values,
%"BSFC", "RES", and "INT_DUR" are vectors whose values correspond to the BMEP
values.
%"res" and "long_dur" are toggles to prevent high residual or to prevent
short durations.

%"Location" is the location of the BMEP value within the range that has the
lowest BSFC
%"idxes" is a vector containing the location of all BMEP values within the
range

loads = [];
idx = 0;

if res==0
    %Determine the BMEP values which fall in the tolerance range
    for i=1:length(BMEP)
        if BMEP(i)<=(100+range)/100*BMEP_ref & BMEP(i)>=(100-
range)/100*BMEP_ref
            idx = idx+1;
            loads(idx) = BMEP(i);
            idxes(idx) = i;
        end
    end
else
    if long_dur>0
    %Determine the BMEP values which fall in the tolerance range but have a
    %residual under 25%
        for i=1:length(BMEP)
            if BMEP(i)<=(100+range)/100*BMEP_ref & BMEP(i)>=(100-
range)/100*BMEP_ref && RES(i) <= 25
                idx = idx+1;
                loads(idx) = BMEP(i);
                idxes(idx) = i;
            end
        end
    else
    %Determine the BMEP values which fall in the tolerance range but have a
    %residual under 25% and a duration over 200 deg CA
        for i=1:length(BMEP)
```

```
            if BMEP(i)<=(100+range)/100*BMEP_ref & BMEP(i)>=(100-
range)/100*BMEP_ref && RES(i) <= 25 && INT_DUR(i) > 200
                idx = idx+1;
                loads(idx) = BMEP(i);
                idxes(idx) = i;
            end
        end
    end
end

%Determine which value and location (if any) results in the minimum BSFC
if isempty(loads)==1
    Location = [];
else
    [val, loc] = min(BSFC(idxes));
    Location = idxes(loc);
end
end
```

References

[1] Woo Sok Chang, *An Electromechanical Valve Drive Incorporating a Nonlinear Mechanical Transformer.*: Doctoral Thesis, Department of Electrical Engineering and Computer Science, Massachusettes Institute Of Technology, 2003.

[2] Yihui Qiu, *Advanced Modeling, Control, and Design of an Electromechanical Engine Valve Drive System with a Limited-Angle Actuator.*: Doctoral Thesis, Department of Electrical Engineering and Computer Science, Massachusettes Institute Of Technology, 2009.

[3] Thomas Dresner and Philip Barkan, "A Review of Variable Valve Timing Benefits and Modes of Operation," *SAE Technical Paper Series*, no. 891676.

[4] Mark A Theobald, Bruno Lequesne, and Rassem Henry, "Control of Engine Load via Electromagnetic Valve Actuators," *SAE Technical Paper Series*, no. 940816.

[5] Stephen Gregory Poulos, *The Effect of Combustion Chamber Geometry on S.I. Engine Combustion Rates -- A Modeling Study.*: Master Thesis, Department of Mechanical Engineering, Massachusettes Institute Of Technology, 1982.

[6] Dennis N. Assanis and Brian K. Bolton, "Variable Valve Timing Strategies For Optimum Engine Perfmormance and Fuel Economy," *The American Society of Mechanical Engineers*, no. 94-ICE-5.

[7] T. Hosaka and M. Hamazaki, "Development of the Variable Valve Timing and Lift (VTEC) Engine for the Honda NSX," *SAE Technical Paper Series*, no. 910008.

[8] Ulrich Kramer and Patrick Phlips, "Phasing Strategy for an Engine with Twin Variable Cam Timing," *SAE Technical Paper Series*, no. 2002-01-1101.

[9] Claus Brüstle and Dietmar Schwarzenthal, "VarioCam Plus – A Highlight of the Porsche 911 Turbo Engine," *SAE Technical Paper Series*, no. 2001-01-0245.

[10] R. Flierl and M. Klüting, "The Third Generation of Valvetrains – New Fully Variable Valvetrains for Throttle-Free Load Control," *SAE Technical Paper Series*, no. 2000-01-1227.

[11] Martin Pischinger, et. al., "Benefits of the Electromechanical Valve Train in Vehicle Operation," *SAE Technical Paper Series*, no. 2000-01-1223.

[12] F. van der Staay, et. al., "Modeling of Exhaust Valve Opening in a Camless Engine," *SAE Technical Paper Series*, no. 2002-01-0376.

[13] Ricardo. (2010, Feb) WAVE Flyer. [Online]. http://www.ricardo.com/Documents/Downloads/Software%20Flyers/Software%20Flyers%20Jan%202010/WAVE_product_flyer_Feb%202010.pdf

[14] Ricardo, SI Tutorial, "tut_si4_16v.wvm", WAVE v8.2.

[15] J. B. Heywood, J. M. Higgins, P. A. Watts, and R. J. and Tabaczynski, "Development and Use of a Cycle Simulation to Predict SI Engine Efficiency and NOx Emissions," *SAE Technical Paper Series*, no. 790291.

[16] and Flynn, P. F. Chen S. K., "Development of Single Cylinder Compression Ignition Research Engine," *SAE Technical Paper Series*, no. 650733.

[17] G. Woschni, "A Universally Applicable Equation for the Instantaneous Heat Transfer Coefficient in the Internal Combustion Engine," *SAE Technical Paper Series*, no. 670931.

[18] John B. Heywood, *Internal Combustion Engine Fundamentals*.: McGraw-Hill, 1988.

[19] Environmental Protection Agency. (2011, March) FTP Driving Schedule. [Online]. http://www.epa.gov/oms/emisslab/methods/ftpcol.txt

[20] Mark Subramaniam, personal communication, October 2010, Manager in Combustion & Flow Simulation at FEV, Inc.

[21] Wai K. Cheng, personal communication, October 2010, Professor of Mechanical Engineering in the MIT Sloan Automotive Laboratory.

[22] Willard W. Pulkrabek, *Engineering Fundamentals of the Internal Combustion Engine*, Second Edition ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2004.

 6551-58